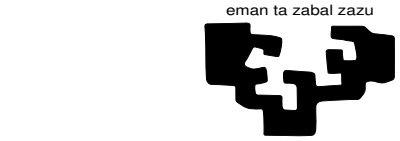

Universidad del País Vasco
Euskal Herriko Unibertsitatea

Efficient meta-heuristics for spacecraft trajectory optimization

by

Abolfazl Shirazi

Supervised by Josu Ceberio and Jose A. Lozano

February, 2021

Konputazio Zientziak eta Adimen Artifizialaren Saila
Departamento de Ciencias de la Computación e Inteligencia Artificial

Efficient meta-heuristics for spacecraft trajectory optimization

by

Abolfazl Shirazi

Supervised by Josu Ceberio and Jose A. Lozano

Dissertation submitted to the Department of Computer Science and Artificial Intelligence of the University of the Basque Country (UPV/EHU) as partial fulfillment of the requirements for the PhD degree in Computer Science

Donostia - San Sebastián, February, 2021

This research was carried out at the Basque Center for Applied Mathematics (BCAM) within the Data Science - Machine Learning group and was supported by La Caixa Foundation Fellowship, the Basque Government through the BERC 2018-2021, Elkartek programs, Spanish Ministry of Economy and Competitiveness MINECO: BCAM Severo Ochoa excellence accreditation SEV-2017-0718, TIN2016-78365R and TIN2017-82626R projects, Spanish Ministry of Science PID2019-1064536A-I00, and by Basque Government consolidated groups 2019-2021 IT1244-19. The support of La Caixa, MINECO and BERC Grants is acknowledged for the four months visit at ISAE-SUPAERO, Toulouse, France.

Acknowledgments

I would like to express my appreciation to my mentors, Josu Ceberio and Jose A. Lozano, for their patient, guidance and constructive advice during my Ph.D program at BCAM and UPV/EHU. My thesis would not have been done without their great support. I also would like to give many thanks to Roberto Armellin, for giving me the opportunity to collaborate as a visiting researcher at ISAE-SUPAERO. This collaboration brought me valuable experiences in my research, along with connections to new people in our research community. Finally, I want to express my deepest appreciation and respect to my family for their love and support during the past few years. I am forever indebted to my parents for giving me the opportunities and experiences that have made me who I am. They selflessly encouraged me to explore new directions in life and seek my own destiny. This journey would not have been possible if not for them, and I dedicate this milestone to them.

RESUMEN

Las metaheurísticas de búsqueda tienen una larga tradición en la informática. Durante los últimos años, diferentes tipos de metaheurísticas, especialmente los algoritmos evolutivos, han recibido una atención notable en la resolución de problemas de optimización del mundo real [2]. Los recientes avances en este campo, junto con el rápido desarrollo de las computadoras de alto rendimiento, hacen posible abordar problemas de optimización en ingeniería, pudiéndose alcanzar resultados desconocidos hasta ese momento. Tras estos rápidos avances, la comunidad científica dirigió su atención hacia el desarrollo de nuevos algoritmos y técnicas para resolver problemas en diferentes áreas de la ciencia y la ingeniería. Entre las diferentes áreas de investigación, la astrodinámica y la ingeniería espacial han sido fuente para el desarrollo de algoritmos evolutivos específicos. Si se examina la cantidad de publicaciones relativas al desarrollo de las metaheurística en las ciencias aeroespaciales, se observa que se dedica un gran número de esfuerzos a desarrollar nuevas técnicas estocásticas y, más concretamente, algoritmos evolutivos innovadores en una variedad de temas. En el último decenio, uno de los problemas más difíciles de la ingeniería espacial, que los investigadores de la comunidad aeroespacial abordan principalmente mediante algoritmos evolutivos novedosos, es la optimización de la trayectoria de las naves espaciales [3].

El problema de la optimización de la trayectoria de las naves espaciales puede describirse simplemente como el descubrimiento de una trayectoria espacial para los satélites y vehículos espaciales que satisfaga algunos criterios. Mientras un vehículo espacial viaja en el espacio para llegar a un destino, ya sea alrededor de la Tierra o de cualquier otro cuerpo celeste, es fundamental mantener o cambiar su trayectoria de vuelo precisamente para llegar al destino final deseado. Esos viajes entre las órbitas espaciales, denominados maniobras orbitales, deben realizarse en condiciones de máxima eficiencia, reduciendo al mínimo algunos objetivos como el consumo de combustible o el tiempo de transferencia. Desde el punto de vista de la ingeniería, la optimización de la trayectoria de las naves espaciales puede describirse como un problema de optimización de caja negra, que puede considerar un mayor o menor número de restricciones, según la formulación del problema.

Para aclarar la motivación principal de la investigación de esta tesis, en primer lugar, es necesario discutir el estado de las tendencias actuales en el desarrollo de los algoritmos evolutivos y abordar los problemas de optimización de la trayectoria de las naves espaciales. En el último decenio se han dedicado a estos temas numerosas investigaciones, principalmente de dos grupos de comunidades científicas. El primer grupo es la comunidad de ingeniería espacial. Un examen general de las publicaciones confirma que el enfoque de los métodos desarrollados en este grupo se refiere principalmente a la modelización matemática y los enfoques numéricos para abordar los problemas de optimización de las trayectorias de las naves espaciales. La mayoría de las estrategias interactúan con conceptos mixtos de métodos semi-analíticos, dis-

cretización, interpolación y técnicas de aproximación. En lo que respecta a la optimización, normalmente se utilizan algoritmos tradicionales y se presta menos atención al desarrollo del algoritmo. En algunos casos, los investigadores tratan de afinar los algoritmos y hacerlos más eficientes. Sin embargo, sus esfuerzos se basan principalmente en el ensayo y el error y en las repeticiones, más que en el análisis de la propia naturaleza del problema de la optimización.

El segundo grupo es la comunidad informática. A diferencia del primer grupo, la mayoría de los esfuerzos en la investigación de este grupo se ha dedicado al desarrollo de algoritmos, en lugar de desarrollar técnicas y enfoques novedosos en la optimización de trayectorias como la interpolación y las técnicas de imitación aproximada. Las investigaciones de este grupo terminan generalmente en algoritmos de optimización muy eficientes y robustos de alto rendimiento. Sin embargo, estos algoritmos no suelen ser evaluados en problemas reales. En su lugar, se seleccionan generalmente los problemas de referencia de optimización estándar para verificar el rendimiento del algoritmo. En particular, cuando se trata de resolver un problema de optimización de la trayectoria de una nave espacial, este grupo trata el problema principalmente como una caja negra con poca concentración en el modelo matemático o las técnicas de aproximación.

Teniendo en cuenta las dos perspectivas de investigación antes mencionadas, puede verse que hay un eslabón perdido entre estos dos esquemas al tratar los problemas de optimización de la trayectoria de las naves espaciales. Por un lado, podemos ver avances notables en los modelos matemáticos y en las técnicas de aproximación sobre este tema, pero sin esfuerzos en los algoritmos de optimización. Por otro lado, hemos desarrollado nuevos algoritmos evolutivos para los problemas de optimización de caja negra, que no aprovechan las nuevas aproximaciones para aumentar la eficiencia del proceso de optimización. En otras palabras, parece que falta una conexión entre las características del problema de la optimización de la trayectoria de las naves espaciales, que controla la forma del dominio de la solución, y los componentes del algoritmo, que controla la eficiencia del proceso de optimización. Esta conexión faltante nos motivó a desarrollar metaheurísticas eficientes para resolver los problemas de optimización de la trayectoria de las naves espaciales.

Al tener el conocimiento sobre el tipo de misión espacial, las características de la maniobra orbital, el modelado matemático de la dinámica del sistema y las características de las técnicas de aproximación empleadas, es posible adaptar el rendimiento de los algoritmos. Conociendo estas características del problema de optimización de la trayectoria de la nave espacial, se puede obtener la forma del espacio de soluciones. En otras palabras, es posible ver cuán sensible es el problema en relación con cada una de sus características. Esta información puede utilizarse para desarrollar algoritmos de optimización eficientes con mecanismos de adaptación, que aprovechen las características del problema para llevar a cabo el proceso de optimización hacia las mejores soluciones. Esa adaptabilidad flexible hace que el algoritmo sea robusto a

cualquier cambio de las características de la misión espacial. Por consiguiente, en la perspectiva del diseño de sistemas espaciales, los algoritmos desarrollados serán instrumentos útiles para obtener trayectorias de transferencia óptimas o casi óptimas dentro del diseño conceptual y preliminar de una nave espacial para una misión espacial.

Teniendo en cuenta esta motivación, el objetivo principal de esta investigación fue el desarrollo de meta-heurísticas eficientes para la optimización de la trayectoria de las naves espaciales. En cuanto al tipo de problema, nos centramos en los problemas de los encuentros espaciales, que cubren la mayoría de las maniobras orbitales, incluyendo los encuentros espaciales de largo y corto alcance. También, en cuanto a la metaheurística, nos concentramos principalmente en los algoritmos evolutivos basados en el modelado probabilístico y la hibridación. Tras la investigación, se han desarrollado dos algoritmos. En primer lugar, se ha desarrollado un algoritmo evolutivo híbrido autoadaptativo para problemas de encuentros espaciales de largo alcance con múltiples impulsos. El algoritmo es un método híbrido, combinado con técnicas de autoajuste y un proceso de refinamiento individual basado en una distribución de probabilidad. Luego, para los problemas de optimización de la trayectoria de los encuentros espaciales de corto alcance, se desarrolla un algoritmo de estimación de distribuciones con mecanismos de conservación de la factibilidad para la optimización continua con restricciones. Los mecanismos propuestos implementan métodos de generación de soluciones iniciales, aprendizaje y mapeo dentro del proceso de optimización. Incluyen mixturas de modelos probabilísticos, algoritmos de detección de valores atípicos y algunas técnicas heurísticas dentro del proceso de la localización de las soluciones factibles.

Paralelamente al desarrollo de los algoritmos, se desarrolla también un programa informático de simulación como aplicación complementaria. Esta herramienta está diseñada para la visualización de los resultados obtenidos de los experimentos de esta investigación. Se ha utilizado principalmente para obtener ilustraciones de alta calidad mientras se simula la trayectoria de la nave espacial dentro de las maniobras orbitales.

ORGANIZACIÓN DE LA TESIS

Esta tesis incluye cuatro capítulos. En primer lugar, en el capítulo 1 se lleva a cabo una revisión bibliográfica sobre la resolución de problemas de optimización de la trayectoria de las naves espaciales. El proceso de resolución se descompone en cuatro pasos clave, que incluyen la modelización matemática del problema, la definición de las funciones objetivo, el desarrollo de un enfoque y la obtención de la solución al problema. Se han descrito varias subcategorías para cada paso. Posteriormente, se han discutido varias clasificaciones y sus características para resolver los problemas.

En el capítulo 2 se presenta un enfoque evolutivo para encontrar el punto de encuentro espacial de largo alcance óptimo en términos de combustible

y tiempo, considerando un impulso limitado. En este enfoque, el problema de Lambert se amplía hacia una transferencia discreta de múltiples impulsos. Aprovechando una solución analítica de transferencia de multi-impulsos, se calcula una solución factible que satisfaga el límite de impulso. A continuación, la solución factible obtenida se utiliza como semilla para generar individuos para un algoritmo evolutivo híbrido autoadaptable para minimizar el tiempo total, sin violar el límite de impulso y manteniendo la masa global de combustible igual o menor que la asociada a la solución analítica. El algoritmo elimina a los individuos similares y los regenera basándose en una combinación de soluciones prometedoras Gaussianas y de distribución uniforme. También se aplican otras mejoras al algoritmo para que sea autoajustable y robusto a las órbitas inicial y final, así como al límite de impulso. Se evalúan varios tipos del algoritmos considerando un conjunto variado de misiones de encuentro. Los resultados revelan que el enfoque puede reducir con éxito el tiempo total de transferencia en las transferencias de múltiples impulsos, minimizando al mismo tiempo la masa de combustible sin violar el límite de impulso. Además, el algoritmo propuesto tiene un rendimiento superior al de los algoritmos evolutivos estándar en términos de convergencia y optimización.

En el capítulo 3, se desarrolla otro enfoque evolutivo incorporando una técnica de control directo para el encuentro espacial de corto alcance en órbitas elípticas con perturbaciones. En este enfoque, los vectores de control se interpolan mediante diversos esquemas de interpolación y se propone un enfoque evolutivo basado en el marco de los algoritmos de estimación de la distribuciones (EDAs por su acrónimo en inglés). El algoritmo desarrollado se beneficia de algunos mecanismos para hacer frente a problemas no lineales. Estos mecanismos están asociados con diferentes etapas de los EDAs, entre ellas la generación de la población inicial, el aprendizaje y la factibilidad. Es posible demostrar que si la población inicial es factible, el algoritmo siempre mantiene poblaciones factibles durante toda la búsqueda. Antes de aplicar el algoritmo desarrollado al problema de optimización de la trayectoria de encuentro en el espacio de corto alcance, se aplica en dos conjuntos de problemas de referencia para la optimización continua con restricciones y su rendimiento se compara con algunos algoritmos y métodos de manejo de restricciones del estado del arte. Los experimentos realizados confirman la velocidad, robustez y la eficiencia del algoritmo propuesto en la solución de diversos problemas con restricciones lineales y no lineales. Tras la experimentación, el algoritmo se aplica en algunos problemas de optimización de la trayectoria en misiones de encuentro de corto alcance. Las simulaciones numéricas muestran que pueden lograrse rápidamente trayectorias de combustible mínimo, satisfaciendo al mismo tiempo las restricciones. El enfoque es robusto para diferentes escenarios y condiciones iniciales. Además, se analiza el rendimiento de los mecanismos de recuperación de la factibilidad empleados y se demuestra que el más eficiente es independiente de la posición relativa inicial y de la velocidad de la nave espacial objetivo. La practicidad del enfoque propuesto se compara con un enfoque de control basado en la función

implícita de Lyapunov. Los resultados muestran la eficiencia y la eficacia del enfoque propuesto para encontrar la solución óptima.

Por último, el capítulo 4 incluye conclusiones y trabajo futuro. Dado que la investigación fue una transacción entre la astrodinámica y la computación evolutiva, el esfuerzo en este capítulo es iluminar las nuevas ideas y logros en esta investigación desde la perspectiva de las comunidades de ingeniería aeroespacial e informática. A este respecto, se revisa el panorama de la investigación en esta disertación, junto con los desafíos y los notables hallazgos. Además, se han señalado varios aspectos en los que la investigación puede continuar en el futuro.

El Apéndice contiene la información relativa al marco de simulación desarrollado en esta investigación. Dado que el desarrollo de este software fue complementario al objetivo principal de la investigación, la introducción de esta plataforma se proporciona en el Apéndice. Este programa informático es un instrumento interactivo para la simulación de la dinámica de los satélites y la orientación autónoma de las naves espaciales. Se proporcionan diferentes modelos matemáticos para la propagación orbital de los satélites de órbita terrestre, que consideran el campo gravitatorio de la tierra con diversas precisiones. El programa informático presentado previamente es una plataforma de visualización tridimensional para la simulación de la órbita espacial con capacidad analítica a través de diversos módulos. Tiene la capacidad de simular trayectorias de transferencia tardía, obtenidas por los algoritmos propuestos, y proporcionar visualizaciones de alta calidad del movimiento y la inclinación de la nave espacial. Aprovechando una interfaz gráfica de usuario, puede evaluar, analizar e ilustrar el movimiento del satélite basándose en diferentes esquemas de propagación orbital y aceleraciones externas. Se simulan varios casos de satélites y naves espaciales autónomas en la misión de encuentro espacial en relación con diferentes modelos de propagación para demostrar el rendimiento de la aplicación de los diferentes modelos, la visualización de la trayectoria en tierra y la optimización de la trayectoria. Los resultados se validan comparándolos con otras herramientas de última generación, como el conjunto de herramientas del sistema AGI (STK).

Contents

List of Figures

List of Tables

List of Algorithms

Preface

Abstract

Meta-heuristics has a long tradition in computer science. During the past few years, different types of meta-heuristics, specially evolutionary algorithms got noticeable attention in dealing with real-world optimization problems [1]. Recent advances in this field along with rapid development of high processing computers, make it possible to tackle various engineering optimization problems with relative ease, omitting the barrier of unknown global optimal solutions due to the complexity of the problems. Following this rapid advancements, scientific communities shifted their attention towards the development of novel algorithms and techniques to satisfy their need in optimization.

Among different research areas, astrodynamics and space engineering witnessed many trends in evolutionary algorithms for various types of problems. By having a look at the amount of publications regarding the development of meta-heuristics in aerospace sciences, it can be seen that a high amount of efforts are dedicated to develop novel stochastic techniques and more specifically, innovative evolutionary algorithms on a variety of subjects. In the past decade, one of the challenging problems in space engineering, which is tackled mainly by novel evolutionary algorithms by the researchers in the aerospace community is spacecraft trajectory optimization [2].

Spacecraft trajectory optimization problem can be simply described as the discovery of a space trajectory for satellites and space vehicles that satisfies some criteria. While a space vehicle travels in space to reach a destination, either around the Earth or any other celestial body, it is crucial to maintain or change its flight path precisely to reach the desired final destination. Such travels between space orbits, called orbital maneuvers, need to be accomplished, while minimizing some objectives such as fuel consumption or the transfer time. In the engineering point of view, spacecraft trajectory optimization can be described as a black-box optimization problem, which can be constrained or unconstrained, depending on the formulation of the problem.

In order to clarify the main motivation of the research in this thesis, first, it is necessary to discuss the status of the current trends in the development of evolutionary algorithms and tackling spacecraft trajectory optimization problems. Over the past decade, numerous research are dedicated to these subjects, mainly from two groups of scientific communities. The first group is the space engineering community. Having an overall look into the publications confirms that the focus in the developed methods in this group is mainly regarding the mathematical modeling and numerical approaches in dealing with spacecraft trajectory optimization problems. The majority of the strategies interact with mixed concepts of semi-analytical methods, discretization, interpolation and approximation techniques. When it comes to optimization, usually traditional algorithms are utilized and less attention is paid to the algorithm development. In some cases, researchers tried to tune the algorithms and make them more efficient. However, their efforts are mainly based on try-and-error and repetitions rather than analyzing the landscape of the optimization problem.

The second group is the computer science community. Unlike the first group, the majority of the efforts in the research from this group has been dedicated to algorithm development, rather than developing novel techniques and approaches in trajectory optimization such as interpolation and approximation techniques. Research in this group generally ends in very efficient and robust optimization algorithms with high performance. However, they failed to put their algorithms in challenge with complex real-world optimization problems, with novel ideas as their model and approach. Instead, usually the standard optimization benchmark problems are selected to verify the algorithm performance. In particular, when it comes to solve a spacecraft trajectory optimization problem, this group mainly treats the problem as a black-box with not much concentration on the mathematical model or the approximation techniques.

Taking into account the two aforementioned research perspectives, it can be seen that there is a missing link between these two schemes in dealing with spacecraft trajectory optimization problems. On one hand, we can see noticeable advances in mathematical models and approximation techniques on this subject, but with no efforts on the optimization algorithms. On the other hand, we have newly developed evolutionary algorithms for black-box optimization problems, which do not take advantage of novel approaches to increase the efficiency of the optimization process. In other words, there seems to be a missing connection between the *characteristics* of the problem in spacecraft trajectory optimization, which controls the shape of the solution domain, and the algorithm *components*, which controls the efficiency of the optimization process. This missing connection motivated us in developing efficient metaheuristics for solving spacecraft trajectory optimization problems.

By having the knowledge about the type of space mission, the features of the orbital maneuver, the mathematical modeling of the system dynamics, and the features of the employed approximation techniques, it is possible to adapt the performance of the algorithms. Knowing these features of the space-

craft trajectory optimization problem, the shape of the solution domain can be realized. In other words, it is possible to see how sensitive the problem is relative to each of its feature. This information can be used to develop efficient optimization algorithms with adaptive mechanisms, which take advantage of the features of the problem to conduct the optimization process toward better solutions. Such flexible adaptiveness, makes the algorithm robust to any changes of the space mission features. Therefore, within the perspective of space system design, the developed algorithms will be useful tools for obtaining optimal or near-optimal transfer trajectories within the conceptual and preliminary design of a spacecraft for a space mission.

Having this motivation, the main goal in this research was the development of efficient meta-heuristics for spacecraft trajectory optimization. Regarding the type of the problem, we focused on space rendezvous problems, which covers the majority of orbital maneuvers, including long-range and short-range space rendezvous. Also, regarding the meta-heuristics, we concentrated mainly on evolutionary algorithms based on probabilistic modeling and hybridization. Following the research, two algorithms have been developed. First, a hybrid self adaptive evolutionary algorithm has been developed for multi-impulse long-range space rendezvous problems. The algorithm is a hybrid method, combined with auto-tuning techniques and an individual refinement procedure based on probabilistic distribution. Then, for the short-range space rendezvous trajectory optimization problems, an estimation of distribution algorithm with feasibility conserving mechanisms for constrained continuous optimization is developed. The proposed mechanisms implement seeding, learning and mapping methods within the optimization process. They include mixtures of probabilistic models, outlier detection algorithms and some heuristic techniques within the mapping process. Parallel to the development of algorithms, a simulation software is also developed as a complementary application. This tool is designed for visualization of the obtained results from the experiments in this research. It has been used mainly to obtain high-quality illustrations while simulating the trajectory of the spacecraft within the orbital maneuvers.

Overview of the Dissertation

This thesis includes four chapters. First, a review for solving spacecraft trajectory optimization problems has been provided in Chapter 1. The solving process is decomposed into four key steps, including mathematical modeling of the problem, defining the objective functions, development of an approach and obtaining the solution of the problem. Several subcategories for each step have been described. Subsequently, important classifications and their characteristics have been discussed for solving the problems.

In Chapter 2, an evolutionary approach is presented for finding the optimal long-range space rendezvous in terms of fuel and time, considering limited impulse. In this approach, the Lambert problem is expanded towards a dis-

cretized multi-impulse transfer. Taking advantage of an analytical form of multi-impulse transfer, a feasible solution that satisfies the impulse limit is calculated. Next, the obtained feasible solution is utilized as a seed for generating individuals for a hybrid self-adaptive evolutionary algorithm to minimize the total time, without violating the impulse limit while keeping the overall fuel mass the same as or less than the one associated with the analytical solution. The algorithm eliminates similar individuals and regenerates them based on a combination of Gaussian and uniform distribution of promising solutions. Other enhancements are also applied to the algorithm to make it auto-tuned and robust to the initial and final orbits as well as the impulse limit. Several types of the proposed algorithm are tested considering varieties of rendezvous missions. Results reveal that the approach can successfully reduce the overall transfer time in the multi-impulse transfers while minimizing the fuel mass without violating the impulse limit. Furthermore, the proposed algorithm has superior performance over standard evolutionary algorithms in terms of convergence and optimality.

In Chapter 3, another evolutionary approach incorporated with a direct control technique is developed for short-range space rendezvous in elliptical orbits with disturbances. In this approach, the control vectors are interpolated via various interpolation schemes and an evolutionary approach based on the framework of Estimation of Distribution Algorithms (EDAs) is propose. The developed algorithm benefits from some mechanisms to deal with non-linear constraints. These mechanisms are associated with different stages of the EDAs, including seeding, learning and mapping. It is shown that besides increasing the quality of the solutions in terms of objective values, the feasibility of the final solutions is guaranteed if an initial population of feasible solutions is seeded to the algorithm. Before the implementation of the developed algorithm into the short range space rendezvous trajectory optimization problem, it is applied on two suites of benchmark problems for constrained continuous optimization and its performance is compared with some state-of-the-art algorithms and constraint handling methods. Conducted experiments confirm the speed, robustness and efficiency of the proposed algorithm in tackling various problems with linear and non-linear constraints. Following the experiments, the algorithm is applied on some trajectory optimization problems in short-range rendezvous missions. Numerical simulations show that minimum-fuel trajectories can be rapidly achieved, while satisfying the terminal constraints, and the approach is robust to different scenarios and initial conditions. Besides, the performance of the employed mapping mechanisms is analyzed and it is shown that the most efficient one is independent of the initial relative position and velocity of the chaser spacecraft. The practicality of the proposed approach is compared with a control approach based on the implicit Lyapunov function. Results show the efficiency and the effectiveness of the proposed approach in finding the optimal solution.

Finally, Chapter 4 includes conclusions and further works. Since the research was a transaction between astrodynamics and evolutionary computa-

tions, the effort in this chapter is to illuminate the novel ideas and achievements in this research from the perspective view of both aerospace engineering and computer science communities. In this regard, overview of the research in this dissertation are reviewed, along with challenges and notable findings. Also, various aspects in which the research can be continued in the future have been brought to attention.

The Appendix contains the information regarding the developed simulation framework in this research. Since the development of this software was complementary to the main research goal, the introduction of this platform is provided in the Appendix. This software is an interactive tool for simulation of satellites dynamics and autonomous spacecraft guidance. Different mathematical models for orbit propagation of Earth-orbiting satellites are provided, which consider Earth's gravitational field with various accuracies. The presented software is a 3D visualization platform for space orbit simulation with analytical capabilities through various modules. It has the ability to simulate transfer trajectories, obtained by the proposed algorithms, and provide high-quality visualizations of the spacecraft motion and attitude. Taking advantage of a graphical user interface, it can evaluate, analyze and illustrate the motion of satellite based on different orbit propagation schemes and external accelerations. Several cases of satellites and autonomous spacecraft in the space rendezvous mission are simulated regarding different propagation models to demonstrate the performance of the application in space mission analysis, ground track visualization and trajectory optimization. Results are validated by comparing with other state-of-the-art tools, such as AGI System Toolkit (STK).

1

Overview of Spacecraft Trajectory Optimization

1.1 Introduction

The spacecraft trajectory optimization problem can be described as the discovery of a trajectory that satisfies some criteria, including initial and terminal conditions. In recent years, considerable progress has been made in the development of methods to find optimal trajectories for spacecraft in various space missions. Within this progress, each step in spacecraft trajectory design can be categorized according to the elements that are involved in finding a solution to the optimal trajectory problem, such as the mathematical model, objective, approach, or, more importantly, the method, technique and algorithm.

Perhaps the first serious attempt to categorize methods for spacecraft trajectory optimization was made by Betts [3] in 1998. The main classification made by Betts considered two famous methods, known as direct and indirect methods, and the primary related techniques in each were summarized. In 2012, Conway [4] made another comprehensive contribution to the numerical approaches applied in dynamical systems. He provided an excellent overview of different methods, similar to Betts' survey, along with practical examples. However, the dynamical systems considered in his survey are in general forms. Other attempts have been also made but limited to specific space missions, such as Earth-Moon trajectories [5], space rendezvous [6], planetary entry [7] and libration points [8], [9]. Different classifications are presented for spacecraft trajectory optimization problems in these researches. Based on the purposes of their taxonomy, each approach or solution has its own advantages and disadvantages [10]. These surveys focused on specific steps of the whole process rather than a general scheme for spacecraft trajectory optimization. Moreover, it is clear from the literature that comparing different taxonomies can be time consuming, although it is generally less complicated than developing one from scratch. While a great deal of research has been done regarding the methods and techniques, an outline that categorizes the key elements within the general process of spacecraft trajectory optimization is missing. This chapter presents such a scheme and it is considered comple-

mentary to all of the previously published survey articles in this subject. It reflects the research that has been done over the past decade while simultaneously providing a roadmap for the general process of spacecraft trajectory optimization.

Before proceeding to the details of the review, it is important to distinguish between several terms in spacecraft trajectory optimization terminology. By looking through the literature, it can be highlighted that a vast number of papers are dedicated to spacecraft trajectory optimization with different terminologies. Approaches, solutions, methods, strategies, techniques and other terms are often used interchangeably. This review also tries to make a distinction between such terms, and uses a clear terminology in order to avoid misunderstandings and confusion when referring to a method, approach, technique or algorithm. It should also be noted that sometimes different parameters are shown with same symbols in literature. Therefore, in this review, every newly introduced parameter in the equations is defined locally in order to avoid misunderstandings with other possible parameters with same symbols.

The entire process of solving a spacecraft trajectory optimization problem can be divided in four steps as depicted in Fig. 1.1. This general process includes mathematical modeling of system dynamics, defining appropriate objectives, developing an approach and, lastly, achieving the solution. Traces of these key elements can be found in textbooks by Betts [11] and Conway [12].

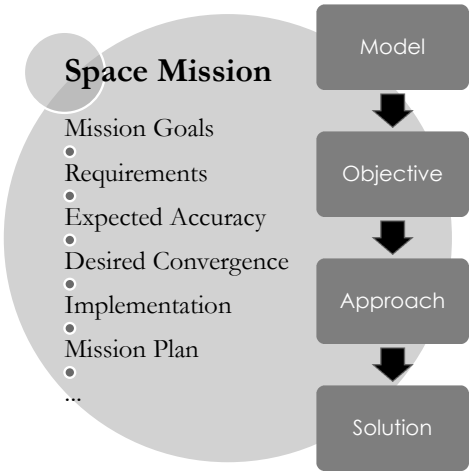

Fig. 1.1: General scheme of spacecraft trajectory optimization process

These steps are represented by *model*, *objective*, *approach* and *solution* respectively. On the other hand, each space mission has several components such as mission requirements, goals, expected accuracy, desired convergence, mission plan, etc. Each of these factors affects the steps of the mentioned

process differently. Therefore, it is important to focus on the taxonomies in each step based on the space mission components in order to make a good decision when choosing a model or employing a method in the spacecraft trajectory optimization process.

The first step to solve not just a spacecraft trajectory optimization problem but indeed any orbital mechanics problem involves a firm understanding of the dynamics inherent in the system. It refers to the mathematical modeling of the problem which involves choosing a set of states to represent the system and derivation of motion equations for spacecraft.

The second step is handling the mission objectives via defining cost functions. Two categories can be considered for this step, one according to the type of the objectives and the other according to the number of objectives.

As the third step, the type of methods and techniques which are dedicated to solving the trajectory design problem are the main feature that characterizes the approach. This step is divided into two categories called analytical and numerical approaches. Analytical approaches are mainly based on the well-known optimal control theory [13]. The purpose of this theory is the determination of a time history of controls that satisfies the physical constraints of the system while minimizing some performance criteria [14] i.e., the cost functions defined in the previous step. There also exist several numerical approaches to solve optimization problems related to space transfers [12]. They fall essentially into two main categories. The first one is called direct methods, which attempt to find the minimum of the cost function by considering state and input vectors. The second one is indirect methods, which involve adjoint equations alongside state and input vectors based on the Pontryagin's Principle [13]. Each one of these two large categories is characterized by both positive and negative aspects, intrinsically limiting their operational fields.

The fourth step is to solve the problem regarding the developed approach. If the analytical approach is developed in the third step, the solution is likely to be a closed form analytical solution. However, if the numerical approach is used, the problem usually turns into a black-box optimization problem and needs numerical algorithms to achieve a solution. Most of the spacecraft trajectory optimization problems end in the latter form, to be solved by numerical techniques rather than by means of an analytical solution. The reason is that a typical spacecraft optimization problem does not have a closed form solution due to its nonlinearity, unless specific conditions and assumptions are considered in the approach. Such assumptions may limit the matching between simulation and reality in spacecraft motion.

This chapter tries to propose a complete taxonomy of spacecraft trajectory optimization problems, along with recently developed concepts and traditional approaches, which covers most of the aspects of this field. The idea is to bring the advantages and disadvantages of various models, objectives, approaches and solutions based on the findings of a considerable number of papers in the literature. In contrast, this review excludes many of the technical details and, instead, provides a road map of currently available tools. General concepts

are briefly described, and references are included for further investigation. In addition, this chapter tries to consolidate seemingly different concepts, methods, and terminology stemming from diverse applications. While a great deal of spacecraft trajectory optimization research has been carried out in the aerospace community, this review attempts to draw from work that has been done in other disciplines as well. It also provides conclusions that can be useful for other disciplines such as applied mathematics and engineering.

This chapter is organized as follows: The next four sections are dedicated to the taxonomies of the four steps mentioned in the process of spacecraft trajectory optimization respectively. Section 1.2 provides the mathematical models required in order to formulate the necessary components of the spacecraft trajectory optimization problem. It outlines several choices of mathematical sets and their corresponding equations of motion according to different categories of space missions. Objective functions in spacecraft trajectory optimization problems, their representation and types are discussed in Section 1.3. Section 1.4 details the approaches used in solving the spacecraft trajectory optimization problem, as well as comparisons of different methods and techniques. Section 1.5 is dedicated to optimization algorithms, including nonlinear programming and metaheuristics. This section aims to taxonomize the optimization algorithms according to space missions and their usage in spacecraft trajectory optimization problems. Section 1.6 summarizes the discussions from this review. It also presents suggestions for future study and new trends in trajectory optimization of spacecraft. Finally, the conclusions are provided in Section 1.7.

1.2 Model

As the first step of facing the spacecraft trajectory optimization problem, the dynamics of the spacecraft need to be mathematically modeled. The spacecraft trajectory model can be referred to a set of ordinary differential equations representing a path or time history of position and velocity of the spacecraft. The equations of motion for the spacecraft which serves as the model can be generally described in first order form as follows [14]:

$$\dot{\mathbf{x}} = f(\mathbf{x}(t), \mathbf{u}(t), t) \tag{1.1}$$

where t represents the time, $\mathbf{x}(t)$ is an n-dimensional time history of the state vector and $\mathbf{u}(t)$ is an m-dimensional time history of the control vector, which serves as the system input. The state vector contains the state variables which can be the position and the velocity vectors of the spacecraft. This general representation is used in the literature as the basic mathematical model for spacecraft trajectory and can be categorized in different aspects and forms, as depicted in Fig. 1.2.

The overall taxonomy of mathematical models of the spacecraft in trajectory optimization problems consists of two minor categories which are *trans-*

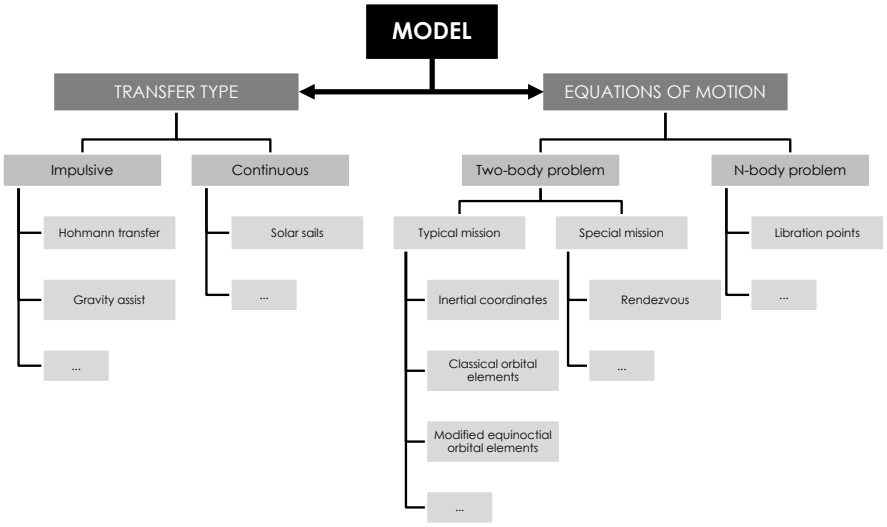

Fig. 1.2: Taxonomy of mathematical models in spacecraft trajectory optimization

fer type and *equations of motion*. While the behavior of the input $\mathbf{u}(t)$ is the matter of interest in the first category, the focus in the second category is on the representation of the whole differential equations as in $f(\mathbf{x}(t), \mathbf{u}(t), t)$. Although other aspects, such as dimensions (2D and 3D), could be also considered in the taxonomy, these two categories are chosen in this taxonomy since they can cover and classify most of the research according to the literature.

1.2.1 Models based on transfer type

In mathematical modeling of the spacecraft trajectory in an orbit transfer, the simulation of the system inputs is an important issue which has a great effect on the trajectory optimization process. Depending on the type of space mission, the model can be either *impulsive* or *continuous*.

1.2.1.1 Impulsive model

Mathematical modeling based on impulsive model is the traditional procedure used to simulate the spacecraft maneuver. In this modeling, the inputs of the system are assumed to be zero $\mathbf{u}(t) = 0$ and the maneuver by the spacecraft is considered as sudden velocity increments ($\Delta v > 0$) with zero burn times ($\Delta t = 0$). If the gravitational acceleration of only one giant mass (for example the Earth) is considered for the problem, the presented differential equation of motion based on state variables in Eq. 1.1 may be reduced

and simplifies into some algebraic equations based on orbital elements. This model is relatively simple to be used in simulation of space trajectory with large accelerations and a rapid spacecraft response to commanded maneuvers. It allows to simulate nearly instantaneous velocity changes necessary for large orbital maneuvers [15]. Impulsive model is typically used when engines with relatively low specific impulse (I_{sp}) and high thrust level are employed. This kind of model, better known as the Kepler model, which was first proposed by Sims and Flanagan [16] to approximate low-thrust trajectories as a series of impulsive Δv's connected by conic arcs, is depicted in Fig. 1.3.

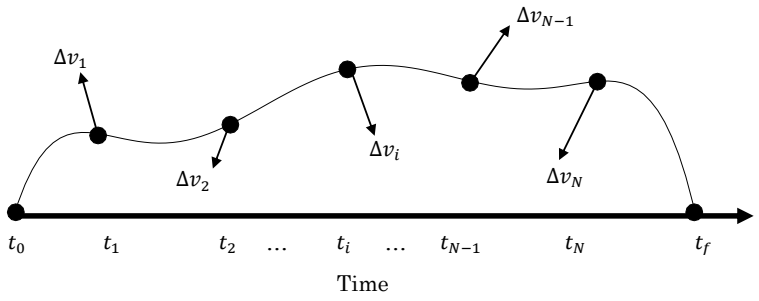

Fig. 1.3: Impulsive discretization scheme

In the impulsive model, a segment (t_i to t_{i+1}) corresponds to an impulsive Δv which can be analyzed by an analytical Kepler propagation with respect to a primary body (Sun, Earth or other planets). Since a closed-form solution is usually known for the state propagation, no numerical integration of the equations of motion is needed, which results in fast computations. The well-known Hohmann transfer [15] is the most practical transfer in which the impulsive model is taken into account for initial design and analysis of orbits in space mission [17].

An advanced concept of impulsive model is called impulsive thrusting. In this model, the trajectory is locally treated as continuous when the engine is on, and the thrust level and the burn time will be considered in the problem. This type of model is sometimes used for optimization of continuous thrust orbit transfers. One example is [18], in which a multi-impulse extended method is proposed for low-thrust trajectory optimization.

As stated, the impulsive model is suitable for space missions with sudden velocity increment. Thrust phases for these missions are typically short compared to the overall mission time, which makes the problem relatively straightforward. As a result, thrust arcs are modelled as isolated, singular events, and the continuous problem can be reduced to a discrete optimization problem which can be represented with the impulsive model. In such cases, the impulsive Δv's readily represents deep space maneuvers, especially for

space missions with several segments (such as gravity assist maneuvers). If no maneuver is needed at the beginning of a segment, the optimizer simply drives the corresponding Δv magnitude to zero. The optimization of the number of impulses, as well as their respective locations, is therefore automatically tackled.

1.2.1.2 Continuous model

The second type for mathematical modeling of the spacecraft is the continuous model. Mathematical models based on this concept are more precise but also more complicated in comparison to impulsive models, since the trajectory is dealt with considering non-zero inputs ($\mathbf{u}(t) \neq 0$). The comparison between impulsive and continuous models is implicitly a mirror of comparing high and low-thrust space missions. From the viewpoint of performance, low-thrust propulsion can improve fuel consumption efficiency for space missions due to their extremely high I_{sp} compared with high-thrust chemical propulsion. However, typical low-thrust trajectories present a major challenge namely the extremely low forces that they generate. Table 1.1 provides detailed characteristics for some specific low and high-thrust propulsion systems [19], [20], [21] [22], [23].

Table 1.1: Characteristics of typical propulsion systems

Propulsion system	Thrust (N)	I_{sp} (s)
Chemical engine	$0.1 - 10^6$	140-460
Cold gas thruster	$0.05 - 200$	50-250
Resisto-jet	$0.002 - 0.1$	150-8000
Arcjet	$0.002 - 0.7$	400-1500
Ion thruster	$1 \times 10^{-5} - 0.2$	1500-5000
Hall thruster	$1 \times 10^{-5} - 1$	1500-6000
Pulsed plasma,thruster	$5 \times 10^{-5} - 0.01$	500-2000
Solar sail	$0.001 - 0.1$	∞

Unlike high-thrust trajectories, the transfer time in low-thrust trajectories is relatively high. Therefore, the continuous model is more adequate for low-thrust trajectories. However, there is some research which deals with the employment of continuous model in analyzing high-thrust transfers as well. For example, in [24], the problem of the optimal space trajectory for the mission to the Apophis asteroid approaching the Earth has been studied with the employment of a continuous model for the departure phase from Earth.

The general representation of continuous transfer is the extension of the Newton equation for the N-body problem as in Eq. 1.2.

$$\ddot{\mathbf{r}} = -G\sum_{i=1}^{n}\frac{m_i(\mathbf{r}-\mathbf{r}_i)}{|\mathbf{r}-\mathbf{r}_i|^3} + \mathbf{\Gamma} \tag{1.2}$$

where $\mathbf{r}$ is the position of the spacecraft, $\mathbf{r_i}$ are the positions of n celestial bodies with masses m_i, and G is the gravitational constant. $\mathbf{\Gamma}$ represents the summation of all accelerations due to sources other than the gravitational force of celestial bodies, such as the space perturbations or the thrust provided by the spacecraft propulsion system [25]. It is clear that, by considering the position vector $\mathbf{r}$ and its time derivative $\dot{\mathbf{r}}$ as the state vectors (i.e., $\mathbf{x}(t) = [\mathbf{r} \quad \dot{\mathbf{r}}]$), Eq. 1.2 will be a specific form of the general model representation as in Eq. 1.1. This is the general equation for any continuous spacecraft trajectory optimization problem. Obviously, by setting $\mathbf{u}(t) = 0$ and considering the maneuvers as sudden velocity increments, the continuous model will turn into the impulsive model. For specific missions and applications such as unperturbed orbits around Earth, the orbit propagation may be simplified to orbital elements. Therefore, the complexity of the model can be changed for various applications, from very simplistic (for example Hohmann transfer) to heavily complicated (e.g., low-thrust interplanetary transfer).

1.2.2 Models based on equations of motion

Besides the concept of a mathematical model for spacecraft trajectory optimization, the representation of the dynamics of the spacecraft motion is a key to categorize the model. Different forms of Eq. 1.2 are considered in the literature depending on the space mission, falling into two main groups of two-body problems and N-body problems. Obviously, this category does not conflict with the previous one. In other words, one can simulate two-body problems or N-body problems with either impulsive or continuous models.

1.2.2.1 Typical two-body problems

The simplest model for spacecraft dynamics is the two-body problem model [15]. This model begins with two point masses and describes their mutual gravitational attraction[17]. In this modeling, the mass of the spacecraft is assumed to be much smaller than the mass of the body it is orbiting. This allows the spacecraft's mass and its gravitational effects on the larger body to be neglected. Moreover, the frame of reference is inertial. This allows for derivatives to be taken without regarding the motion of the reference frame. Besides, both the celestial body and the spacecraft are supposed to be point masses and no other forces are applied to either body [26]. These assumptions allow for the basic formulation of the two-body problem, however they constitute an imperfect model.

The most common mathematical model of spacecraft dynamics regarding the mentioned assumptions for typical two-body problems can be described as the well-known non-Keplerian two-body problem equation [15], [26]:

$$\ddot{\mathbf{r}} = -\frac{\mu}{r^3}\mathbf{r} + \boldsymbol{\gamma} \tag{1.3}$$

This equation of motion is best described initially using an independent inertial coordinate frame. In this equation, $\mathbf{r}$ denotes the position of the spacecraft relative to inertial coordinate system, μ is the gravitational constant of the central mass, and $\boldsymbol{\gamma}$ is the acceleration due to engine thrust. Rewriting this equation in scalar form yields the following set of first-order derivatives [17]:

$$\begin{pmatrix} \dot{r}_x \\ \dot{r}_y \\ \dot{r}_z \\ \dot{v}_x \\ \dot{v}_y \\ \dot{v}_z \end{pmatrix} = \begin{pmatrix} v_x \\ v_y \\ v_z \\ -\frac{\mu}{r^3} r_x + \gamma_x \\ -\frac{\mu}{r^3} r_y + \gamma_y \\ -\frac{\mu}{r^3} r_z + \gamma_z \end{pmatrix} \tag{1.4}$$

where r_x,r_y,r_z are the position components ($r = r_x\mathbf{i} + r_y\mathbf{j} + r_z\mathbf{k}$), v_x,v_y,v_z are the velocity components ($\mathbf{r} = v_x\mathbf{i} + v_y\mathbf{j} + v_z\mathbf{k}$) and γ_x,γ_y,γ_z are the acceleration components ($\boldsymbol{\gamma} = \gamma_x\mathbf{i} + \gamma_y\mathbf{j} + \gamma_z\mathbf{k}$) in Earth Centered Inertial (ECI) frame, where $\mathbf{i}$, $\mathbf{j}$, and $\mathbf{k}$ denote the unit vectors of the coordinate system attached to the frame of reference.

Besides Cartesian form, cylindrical coordinates are sometimes considered in research as follows.

$$\begin{pmatrix} \ddot{r} - r\dot{\theta}^2 + \frac{\mu}{s^3} r \\ r\ddot{\theta} + 2\dot{r}\dot{\theta} \\ \ddot{z} + \frac{\mu}{s^3} r \end{pmatrix} = \begin{pmatrix} \gamma_r \\ \gamma_\theta \\ \gamma_z \end{pmatrix} \tag{1.5}$$

where $s = \sqrt{r^2 + z^2}$, and γ_r,γ_θ,γ_z are the acceleration components in cylindrical coordinate systems. These general trajectory equations of motion are vastly used in many spacecraft trajectory optimization problems [27, 28], specifically for analyzing perturbed orbits [29] and low-thrust transfers [30]. Although the Cartesian and cylindrical forms are often used for typical spacecraft trajectory optimization problems [31], other forms based on the variation of parameters are sometimes used in spacecraft trajectory optimization.

Another form of mathematical model for spacecraft trajectory optimization is in terms of classical orbit elements. The six classical orbital elements [26] can be derived from the position and velocity vectors directly [15]. Nevertheless, sometimes the following Lagrange equations are used in mathematical modeling of spacecraft dynamics [26]:

$$\frac{da}{dt} = \Big(\frac{2e\sin\theta}{n\sqrt{1-e^2}}\Big)\gamma_r + \Big(\frac{2a\sqrt{1-e^2}}{nr}\Big)\gamma_t \tag{1.6}$$

$$\frac{de}{dt} = \Big(\frac{\sqrt{1-e^2}\sin\theta}{na}\Big)\gamma_r + \Big(\frac{\sqrt{1-e^2}}{na^2e}\big(\frac{a^2(1-e^2)}{r} - r\big)\Big)\gamma_t \tag{1.7}$$

$$\frac{di}{dt} = \Big(\frac{r\cos(\omega+\theta)}{na^2\sqrt{1-e^2}}\Big)\gamma_n \tag{1.8}$$

$$\frac{d\Omega}{dt} = \Big(\frac{r\sin(\omega+\theta)}{na^2\sqrt{1-e^2}\sin i}\Big)\gamma_n \tag{1.9}$$

$$\frac{d\omega}{dt} = \Big(-\frac{\sqrt{1-e^2}\cos\theta}{nae}\Big)\gamma_r + \Big(\frac{\sqrt{1-e^2}}{nae}\big(1+\frac{r}{a(1-e^2)}\big)\sin\theta\Big)\gamma_t - \Big(\frac{r\cot i\sin(\omega+\theta)}{na^2\sqrt{1-e^2}}\Big)\gamma_n \tag{1.10}$$

$$\frac{dM}{dt} = \Big(\frac{(1-e^2)\cos\theta}{nae} - \frac{2r}{na^2}\Big)\gamma_r - \Big(\frac{1-e^2}{nae}\big(1+\frac{r}{a(1-e^2)}\big)\sin\theta\Big)\gamma_t + n \tag{1.11}$$

where the classical orbital elements, a, e, i, Ω, ω, M are semi-major axis, eccentricity, inclination, right ascension of ascending node (RAAN), argument of perigee and mean anomaly. n is the mean motion, defined as $n = \sqrt{\frac{\mu}{a^3}}$ and θ is the true anomaly. The parameters γ_r, γ_t, γ_n denote radial, tangential, and normal components, respectively [32].

The advantage of using this set of equations is that they provide the variation of classical orbital elements directly from the acceleration without the need for state variables. This method of defining an orbital state is intuitive but unfortunately has a number of singularities that tend to complicate the equations of motion. For instance, at zero inclination the right ascension of ascending node loses meaning. Similarly, for zero eccentricity the argument of perigee becomes indistinguishable from the true anomaly. These singularities can be clearly seen in their equations of motion. Due to the existence of these singularities, the classical orbital elements are not necessarily the best set of states for numerical analysis.

The other model of completely defining an orbit is by the use of the modified equinoctial orbital elements. This element set maintains the mathematical advantages of the classical orbital elements without going singular for circular or prograde equatorial orbits. The set of differential equations defining the spacecraft dynamics based on equinoctial orbital elements is as follows [11]:

$$\frac{dp}{dt} = 2\sqrt{\frac{p^3}{\mu}}\frac{1}{W}f_N \tag{1.12}$$

$$\frac{df}{dt} = \sqrt{\frac{p}{\mu}}\frac{1}{W}\big(W\sin(L)f_s + A(L)f_N - g(h\sin(L) - k\cos(L))f_W\big) \tag{1.13}$$

$$\frac{dg}{dt} = \sqrt{\frac{p}{\mu}}\frac{1}{W}\big(-W\cos(L)f_s + B(L)f_N + f(h\sin(L) - k\cos(L))f_W\big) \tag{1.14}$$

$$\frac{dh}{dt} = \frac{1}{2}\sqrt{\frac{p}{\mu}}\frac{X}{W}\cos(L)f_W \tag{1.15}$$

$$\frac{dk}{dt} = \frac{1}{2}\sqrt{\frac{p}{\mu}}\frac{X}{W}\sin(L)f_W \tag{1.16}$$

$$\frac{dL}{dt} = \sqrt{\frac{\mu}{p^3}}W^2 + \sqrt{\frac{p}{\mu}}\frac{1}{W}\big(h\sin(L) - k\cos(L)\big)f_W \tag{1.17}$$

where the following abbreviations have been used:

$$s = \sqrt{1 - f^2 - g^2} \tag{1.18}$$

$$X = 1 + h^2 + k^2 \tag{1.19}$$

$$W = 1 + f\cos(L) + g\sin(L) \tag{1.20}$$

$$A(L) = f + \cos(L)(1 + W) \tag{1.21}$$

$$B(L) = g + \sin(L)(1 + W) \tag{1.22}$$

where p, f, g, h, k and L are the modified equinoctial orbital elements. Also, f_N, f_S, and f_W are components of the perturbing acceleration in the directions perpendicular to the radius vector in the direction of motion, along the outward radius vector, and normal to the orbital plane in the direction of the angular momentum vector, respectively. This set of equations is employed in a lot of research with various space missions including interplanetary transfers with gravity assist maneuvers [33], [34]. Besides, the use of modified equinoctial elements to describe the osculating orbits is a good choice due to the easy formulation and robustness to uncertainties. See [35] for an instance regarding these characteristics.

Other forms of differential equations may be used as the model for typical two-body problems in spacecraft trajectory optimization. To be more specific, there are twenty two identified candidate orbit element sets plus variations defined in terms of Euler angles, Euler parameters, functions of classical elements, quaternions, set-III elements, fast or slow variables, or canonical variables. These other forms of orbital elements are well explained in a survey by Hintz [36]. However, the model presented by the equation of state vectors including position and velocity is used more frequently in spacecraft trajectory optimization problems. The three sets of equations for modeling the two-body problems are compared in Table 1.2.

As it has been demonstrated, when external acceleration is introduced, time variation of classical orbital elements can be calculated based on the

Table 1.2: Comparison of dynamic models for continuous thrust transfers

	Inertial coordinates	Classical orbital elements	Modified equinoctial orbital elements
Having physical meaning	Normal	High	Low
Extending to other forms	Easy	Hard	Hard
Suffering from singularities	No	Yes	No
Practicality for numerical averaging	No	No	Yes
Complexity of equations	Low	High	Medium

standard variation of physical parameters. Euler angles are used to parameterize the orientation of the orbit plane. However, due to the inherent singularities of the Euler angles, the variational equations may become singular for zero eccentricity and/or zero inclination because Ω and ω are indeterminate for $i = 0$, or π and ω is indeterminate for $e = 0$.

While the modified equinoctial orbital elements avoid the singularities of the classical orbital elements, the main disadvantage of using them is that from direct inspection it is not intuitively obvious what is happening physically to the system. The classical orbital elements directly relate to the physical geometry of the orbit and are much simpler to directly interpret than the equinoctial orbital elements.

There is significant freedom in the choice of a suitable set of state variables or elements. The modified equinoctial element set is the only one that is non-singular for all values of eccentricity and inclination. This set also employs elements that are not far from the classical ones, so that transforming and interpreting them in terms of physically significant parameters is relatively easier than using classical orbital elements. Therefore, it is advisable to use the modified equinoctial orbit element set for the research and technology development task. They can also be used for the integration of orbits with special and general perturbations, as well as differential corrections in orbit determination. However, the other orbit element sets could prove to be convenient in specific applications where the singularities are not a problem [36].

1.2.2.2 Rendezvous

Besides the models for typical two-body problems, the general equation of motion can be reformulated and turned into new representations regarding any special space missions. One of the challenging space missions in literature is the space rendezvous. Rendezvous in space between two spacecraft is accomplished when both space vehicles attain the same position vector and velocity vector at the same time. However, at the time the rendezvous sequence is initiated, they may be very far apart, possibly with one satellite at liftoff. The first part of a rendezvous sequence is the phasing step, which is to perform the maneuvers in the timing sequence that will bring the two satellites into close

proximity. For this step, the state vectors model which includes the position and velocity [26] is usually valid and selected as the general dynamic equation for describing the spacecraft motion.

The next step is the terminal rendezvous. It performs the maneuvers that induce the relative motion between the spacecraft that is required for rendezvous and docking, i.e., the motion of one spacecraft (chaser or active vehicle) with respect to the other (target or passive vehicle). The coordinate frame is attached to one (target) of the satellites in this maneuver [37]. The most used model for this mission in spacecraft rendezvous is given by Clohessy-Wiltshire equations, which have been widely adopted to study the spacecraft relative motion problems. By assuming small distance between the chaser and the target, the linearized equations of the relative motion between them can be described as below [15], [37]:

$$\begin{pmatrix} \ddot{x} - 2n\dot{y} - 3n^2 x \\ \ddot{y} + 2n\dot{x} \\ \ddot{z} + n^2 z \end{pmatrix} = \begin{pmatrix} \gamma_x \\ \gamma_y \\ \gamma_z \end{pmatrix} \tag{1.23}$$

where n denotes the mean motion of the target vehicle. These equations can be used to study the forces required to perform an orbit rendezvous, the displacements from a reference trajectory produced by maneuvers or other velocity changes and the effects of perturbations on the displacements from a reference trajectory. These second-order differential equations are valid for small displacements (a few tens of kilometers in the radial and out-of-plane directions) but remain correct for an order of magnitude (hundreds of kilometer) of larger change in the down track coordinate. Several articles can be referred to as samples of rendezvous missions for additional information [38], [39].

The Clohessy-Wiltshire equation set is derived from the assumptions that the two spacecraft run on neighboring two-body circular orbits and the relative distance between the two spacecraft is much shorter than their geocentric distance. Moreover, first-order approximations are used so that second- and higher-order terms of relative positions and velocities are ignored. It needs improvements in order to describe relative trajectories not satisfying these assumptions. For the sake of brevity, the detailed description of improved relative dynamics equations for space rendezvous is omitted here and the reader is referred to the survey by Luo et al. in 2014 [6].

1.2.2.3 Libration points

Libration points, sometimes referred as Lagrange's points, are essentially the gravitational equilibrium in celestial mechanics, where a spacecraft is able to keep stationary with respect to the primary and secondary bodies without fuel consumption. Therefore, they can motivate numerous space missions due to their special locations. Simulation of transfers to these points needs

models based on N-body problem since the gravitational force of more than one celestial body is considered on the spacecraft. The trajectories used by these missions are solutions of the Circular Restricted Three-Body Problem (CR3BP). CR3BP is the simplest model to study the three-body problem, as well as the most useful one to investigate the motions and phase space structure near libration points. However, it is sometimes not accurate enough for astronautical applications.

The planar CR3BP describes the motion of a spacecraft moving in the gravitational field of two primaries P_1 and P_2, with masses m_1 and m_2. The equations of motion in the normalized synodic reference frame, are [26]:

$$\begin{pmatrix} \ddot{x} - 2\dot{y} \\ \ddot{y} + 2\dot{x} \end{pmatrix} = \begin{pmatrix} \Omega_x \\ \Omega_y \end{pmatrix} \tag{1.24}$$

with the effective potential given by

$$\Omega(x, y) = \frac{1}{2}(x^2 + y^2) + \frac{1-\mu}{r_1} + \frac{\mu}{r_2} + \frac{\mu(1-\mu)}{2} \tag{1.25}$$

and

$$\mu = \frac{m_2}{m_1 + m_2} (m_2 > m_1) \tag{1.26}$$

$$r_1 = \sqrt{(x-\mu)^2 + y^2} \tag{1.27}$$

$$r_2 = \sqrt{(x+1-\mu)^2 + y^2} \tag{1.28}$$

where r_1 and r_2 denote the distances from the particle to P_1 and P_2, respectively, and μ (not to be confused with the gravitational constant in previous subsections), known as the mass parameter of the CR3BP, is the dimensionless mass of P_2. The normalized variables are such that the distance between P_1 and P_2, the sum of their masses, and their angular velocity around the barycenter are normalized to one. So, one complete rotation of the primaries around their barycenter with respect to an inertial frame occurs in 2π dimensionless units of time, and, in the synodic frame, P_1 and P_2 are fixed at $(\mu, 0)$ and $(\mu - 1, 0)$, respectively. Halo orbits located around the collinear libration points in the CR3BP can be well established regarding this modeling. See [40] and [41] for some instances.

The presented dynamic equations are suitable for the 2D problem. The 3D form of CR3BP has one additional differential equation for Z axis. Besides, the center of the main coordinate system may be modified and shifted toward different masses in some research. The general 3D representation of CR3BP is as follows:

$$\begin{pmatrix} \ddot{x} - 2\dot{y} - x \\ \ddot{y} + 2\dot{x} - y \\ \ddot{z} \end{pmatrix} = \begin{pmatrix} -\dfrac{(1-\mu)(x+\mu)}{r_1^3} - \dfrac{\mu(x-1+\mu)}{r_2^3} \\ -\dfrac{(1-\mu)y}{r_1^3} - \dfrac{\mu y}{r_2^3} \\ -\dfrac{(1-\mu)z}{r_1^3} - \dfrac{\mu z}{r_2^3} \end{pmatrix} + \begin{pmatrix} \gamma_x \\ \gamma_y \\ \gamma_z \end{pmatrix} \tag{1.29}$$

Numerous researches use this model for libration points [42]. The more general representation of system dynamics for Lagrange's points is the extended continuous thrust form for N-body problem in the Earth-centered inertial coordinate frame as [15]:

$$\ddot{\mathbf{r}} = -\frac{\mu_E}{r^3} + \sum_{i=1}^{N} \mu_i \left(\frac{\mathbf{r_i} - \mathbf{r}}{|r_i - r|^3} - \frac{\mathbf{r_i}}{r_i^3} \right) + \mathbf{\Gamma} \tag{1.30}$$

where μ_i is the gravitational constants for any of the planets considered in the model, while $\mathbf{r_i}$ is the position vector for that planet. An example of this model is used in [43], which considers the solar radiation pressure as the perturbed acceleration.

Other forms of equations describing the dynamics of Lagrange's points vary for different coordinate systems and expected accuracies. One representation is the Elliptical Four-body Problem (EFBP) in which the Moon moves around the Earth in an elliptical motion and the Earth-Moon system moves around the Sun in a circular orbit. All the central bodies are in the same plane. Although the EFBP is not the most faithful model for the spacecraft, it does include the most important facts, the Sun's direct influence and the Moon's elliptical motion, which cannot be analyzed in the Circular Restrict Three-Body Problem (CR3BP). More details about mathematical models for libration points can be found in [44].

1.2.3 Summary

In spacecraft trajectory optimization, modeling the engineered system dynamics is the primary step. The model of the dynamic system is a set of equations (differential equations) that represents the dynamics of the system using laws of physics. The model allows the study of spacecraft transients and steady state performance. As the model becomes more detailed, it also can become more accurate. Model accuracy needed for spacecraft trajectory optimization is normally simpler than the model used for system simulation. Ignoring some physical phenomena, linearly approximating nonlinear characteristics and using the approximation of lumped parameters in spacecraft are the ways which turn complex models to simple ones for trajectory optimization.

1.3 Objective

The second key element of the spacecraft trajectory optimization process is defining objectives based on the space mission requirements. Objectives are defined by means of some functions, usually referred to cost functions in optimal control terminology or objective functions in computer science terminology. They may comprise fuel mass, total velocity increment, state errors, transfer time, or acceleration. Besides, other components may be also considered in

some space missions due to the employment of specific techniques or mission criteria.

Although the objective functions vary from problem to problem, its general form, better known as Bolza cost function [13], can be defined as follows [14]:

$$J(\mathbf{x}, \mathbf{u}, t) = h(\mathbf{x}(t_f), t_f) + \int_{t_0}^{t_f} g(\mathbf{x}(t), \mathbf{u}(t), t)dt \tag{1.31}$$

where t_0 and t_f are the initial and final times, respectively. The function h represents the Mayer term which denotes the cost related to the final states and g is referred to as the Lagrange term or the running cost which tracks the state and control costs that occur through their entire time histories. This general form of the objective function represents a complete cost function in optimal control theory. The objective function may contain just the Mayer term, just the Lagrange term, or both, depending on what is being optimized in space travel. Various forms of objective function can be categorized in two different aspects, including *type* and *quantity* as illustrated in Fig. 1.4.

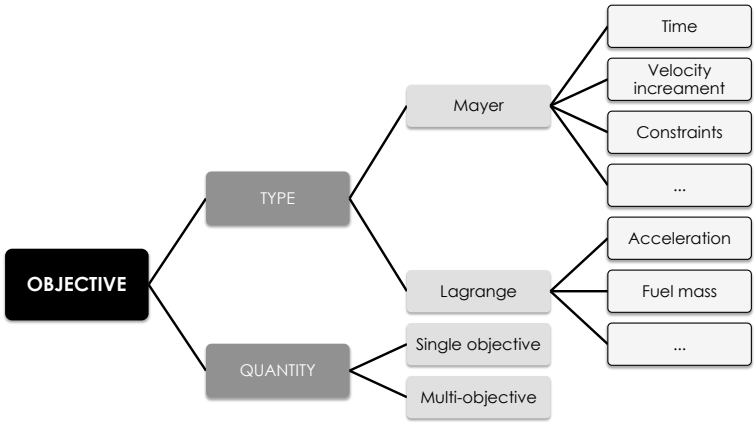

Fig. 1.4: Taxonomy of objectives in spacecraft trajectory optimization

One common and familiar definition between all types of objective functions is presented by Conway [4]. It states that the objective of any space mission depends on two concepts, including having minimum time or minimizing control effort, referring relatively to the Mayer and Lagrange terms in Eq. 1.31. This type of definition can be considered as a subcategory for the type of the objectives. As for the number of objectives, cost functions can be roughly divided into single objective and multi-objective ones [45]. As will be discussed in this section, the simplest way to deal with multipurpose spacecraft optimization problem is to consider an overall objective function being the weighted sum of the single objective functions [46].

The following subsections are dedicated to the different types of objective functions according to the taxonomy, ending with a brief discussion about the typical techniques for dealing with multi-objective problems.

1.3.1 Mayer

The first types of objectives are those which are related to the Mayer term. These objectives are functions of state variables at the end of a transfer trajectory.

1.3.1.1 Time

The time in spacecraft trajectory optimization problems is simply the transfer time in space travels. While the control effort has different kinds of representations as previously mentioned, such as fuel mass, thrust level or acceleration, the time has a simple representation in most of the problems. For minimum time problems, the cost function can be simply defined as:

$$J = t_f \tag{1.32}$$

where t_f is the transfer time. One example regarding the use of this cost function is the trajectory optimization of libration points in Earth-Moon system, in which the duration of flight to perform the mission purposes is expected to be minimized [47]. Wang et al. [48] used this cost function to minimize the transfer time for a Earth to Mercury space mission utilizing solar sails.

1.3.1.2 Velocity increment

In many spacecraft trajectory optimization problems the objective is to minimize the velocity increment or the summation of the increments in multiple phases. A typical example is the well-known multiple gravity assist mission with n stages [49]. In this problem each impulse causes a mass consumption proportional to the modulus of the change of velocity. Therefore, in order to minimize the overall mass consumption, the following objective function is usually considered:

$$J = \sum_{i=0}^{n} |\Delta v_i| \tag{1.33}$$

where Δv_i is the single change of velocity dedicated to each trajectory between two astronomical body (planets or asteroids). This representation of cost function is very popular specifically for multi-impulse problems [50], [18] and even rendezvous missions [38], [39].

1.3.1.3 Initial and terminal conditions

Although the terminal conditions (or even in some cases the initial conditions) are generally considered as constraints, there are some researches which deal with these constraints as objectives [51]. The general form will be as follows:

$$J = \phi(\mathbf{x}(t_0), \mathbf{x}(t_f)) \tag{1.34}$$

where ϕ is the initial and terminal constraint. Note that this function is more general than the function h given at Eq. 1.31. This kind of cost function is used in several articles including gravity assist maneuvers [33], [34], continuous interplanetary trajectories [48], spacecraft reentry [52] and orbit rising mission [53].

1.3.2 Lagrange

Unlike the objectives related to the Mayer term, the second types of objectives are functions in the form of Lagrange term in Eq. 1.31. These costs are integrals of inputs or state variables in the transfer trajectory.

1.3.2.1 Acceleration

One cost function which is usually considered to be minimized in space missions is the integration of the square of spacecraft acceleration within the transfer trajectory. Its representation is as follows:

$$J = \frac{1}{2}\int_{t_0}^{t_f} \gamma^2 dt \tag{1.35}$$

where γ is the magnitude of the spacecraft acceleration, typically due to the propulsion system, while t_0 and t_f are the initial and final time of spaceflight. This representation is popular in researches including thrust minimization [54], [55]. Other forms of cost functions for acceleration are popular in spacecraft trajectory optimization problems, specifically in shape-based techniques [56]. The reason is that in these researches, the state vectors are directly interpolated via polynomials with discretization [43]. So the acceleration (similarly the thrust magnitude) will be achieved as a function of optimization variables. In such cases, the total velocity increment can be calculated by integrating the norm of the acceleration vector as:

$$J = \Delta v = \int_{t_0}^{t_f} \sqrt{{\gamma_x}^2 + {\gamma_y}^2 + {\gamma_z}^2} dt \tag{1.36}$$

where γ_x, γ_y, γ_z are the components of overall acceleration of the spacecraft within the transfer trajectory. Obviously, these components can be substituted by any coordinate system [57]. In some researches [58], acceleration is sometimes represented by T/m as follows:

$$J = \int_{t_0}^{t_f} \frac{T}{m} dt \tag{1.37}$$

where T and m are the thrust magnitude and the mass of the spacecraft, respectively. In this case, the maximum value or the integral of thrust magnitude will be considered as the cost function. Eq. 1.36 along with Eq. 1.33 are both representations of velocity increment in discrete and continuous domains, respectively. The type of the model and the approach determine which one should be used in spacecraft trajectory optimization.

1.3.2.2 Fuel mass

Besides velocity increment and acceleration, sometimes fuel mass is considered as an alternative representation of energy. The study of fuel cost function for spacecraft trajectories has been taken into consideration for a long time [59]. One representation is as below.

$$J = \int_{t_0}^{t_f} \dot{m}_f dt \tag{1.38}$$

where m_f is the fuel mass of the spacecraft within the transfer trajectory. If the transfer time and the mass decreasing rate is fixed, the fuel mass will be independent of the transfer trajectory and can be calculated directly. In such cases, the fuel mass takes the Mayer form simply as $J = m_f$, as in [27]. Also, in some research the fuel mass itself is an input of the optimization process which turns this variable into a known and fixed parameter in the overall process.

1.3.3 Other objectives

Besides the common objectives, sometimes additional costs are considered in some problems, depending on the approach or mission criteria. For example, Luo and Tang [60] employed the following cost function in order to prevent the solution from having a non-smooth trajectory:

$$J = \frac{1}{2} \int_{t_0}^{t_f} [\mathbf{x}(t)]^T Q [\mathbf{x}(t)] dt \tag{1.39}$$

where Q is the state weighting coefficient. Employment of this cost function, besides other familiar cost functions, overcomes the production of a non-smooth or badly scaled trajectories.

Optimization of multi-spacecraft constellations problem by Li et al. [61] is another example of employing a special cost function. Since the problem in this research is formation flying, a very specific cost function including several terms is considered. Objectives such as collision, path length, execution time, fuel consumption and fuel distribution are considered simultaneously.

As another example in the multi gravity assist trajectory optimization problem, the constraints on the minimum distance from the center of each planet can be included in the objective function through appropriate penalty terms [62]. Therefore, some modifications are applied to the objective function. Vasile and Locatelli [49] considered the following modification to the typical cost function in that problem:

$$J = J_0 + \sum_{i=1}^{n} w_i \max[r_i - \gamma_i] \tag{1.40}$$

where J_0 is a typical cost function in the form of Eq. 1.33, w_i are given penalty parameter values and γ_i are the pericentre radius of planets in n stages of the space mission. This form of objective function has also been used in [63].

1.3.4 Scalarization

As stated, objectives can also be categorized according to their number. Single-purpose methods attempt to determine the solution taking into account a single criterion, whereas multi-purpose methods search for a trade-off among several distinct criteria. Almost every advanced spacecraft trajectory optimization problem involves multiple conflicting criteria and it is not possible to mathematically define a single optimal solution. However, a set of compromises called Pareto optimal solutions can be defined. In such problems, there is no unique solution that optimizes both objectives [4]. For instance, in order to force the optimization to meet the boundary conditions while searching the proper set of inputs for the maximization or minimization of the given cost function, it is convenient to attempt to minimize an augmented cost function. This process, better known as scalarization [64], is the method which transforms the problem into a single-objective optimization one involving possibly some parameters or additional constraints. Separate terms in the cost function are given appropriate weights designating their relative importance in the optimization. This is perhaps the most difficult part of designing the cost function. There are an infinite number of weighting combinations if multiple terms are present. One representation of using a weighting coefficient in a general form is presented in [65] as the following:

$$J(\mathbf{x}, \mathbf{u}, t) = h(\mathbf{x}(t_f), t_f) + \alpha \int_{t_0}^{t_f} g(\mathbf{x}(t), \mathbf{u}(t), t) dt \tag{1.41}$$

where α denotes the relative importance of minimizing cost to terminal conditions. The primary purpose of this weighting factor is to balance the cost function such that the Mayer and Lagrange terms have the same relative order of magnitude. It has been stated that for the impulsive cases where the thrust time is small relative to the scenario time, it required a weighting factor on the order of 10^{-2}. For the continuous case the thrust time was larger relative

to the total scenario time requiring this weighting factor to decrease to the order of 10^{-6} [65]. However, each scenario requires specific manipulation of this variable in order to properly balance the cost function.

The rest of the methods in scalarization techniques are more practical than Eq. 1.41. In these techniques, the Mayer and Lagrange terms in the main equation are separated into several minor cost functions J_1,J_2,... J_p for p number of minor cost functions. It allows the user to specify preferences, which may be articulated in terms of goals or the relative importance of different objectives [46]. Most of these methods incorporate parameters, which are coefficients, exponents, constraint limits, etc, that can either be set to reflect decision-maker preferences, or be continuously altered in an effort to represent the complete Pareto optimal set. Some forms of modified cost functions in spacecraft trajectory optimization problems are as follows:

$$J = \sum_{i=1}^{n} J_i \tag{1.42}$$

$$J = \sum_{i=1}^{n} \alpha_i J_i \tag{1.43}$$

$$J = \sum_{i=1}^{n} \frac{J_i}{\alpha_i} \tag{1.44}$$

$$J = \sum_{i=1}^{n} J_i^2 \tag{1.45}$$

where in some of them each cost function is multiplied or divided by a weighting coefficient.

Choosing proper weighting coefficients in this form is a challenging issue and has been tackled in different ways in various researches [66]. For instance, Bolle and Circi [67] tackled this problem by choosing the maximum propagation time allowed in the simulation as the related coefficient for the time cost function and manually tuning the other weighting coefficient (magnitude of the error tolerance permitted during the mission design phase) in order to make both cost functions behave in the same order. As such, extreme care must be taken in properly balancing the relative weights in the cost function. For fundamental background in the associated multi-objective optimization, the reader should refer to Marler and Arora [45].

1.4 Approach

Hitherto, the spacecraft dynamics and the objectives have been modeled and properly defined for the spacecraft trajectory optimization problem. The problem can now be introduced as a general representation as depicted in Fig. 1.5.

Details for the dimensions of variables are provided in [68]. The next step in spacecraft trajectory optimization is to develop an approach for finding the optimal trajectory. Since this step is a vast subject, only an overview of approaches with a brief discussion is provided in this section. For a fundamental background in the associated approaches, the reader should refer to [12].

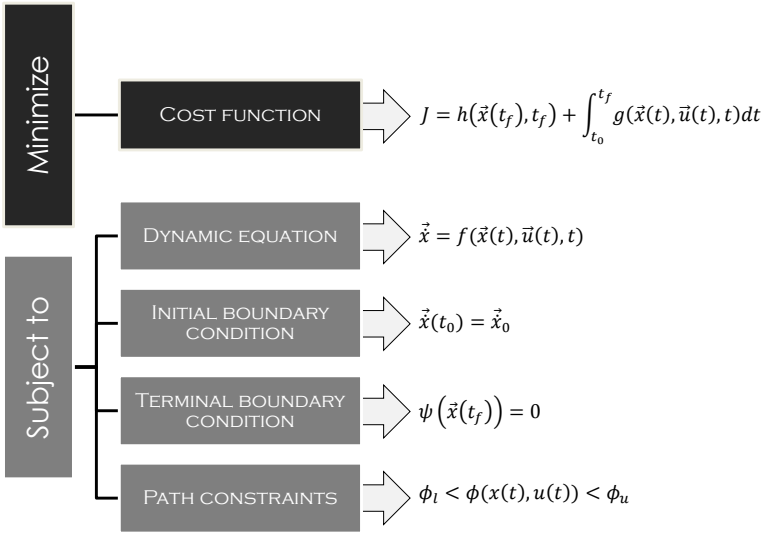

Fig. 1.5: Mathematical representation of a spacecraft trajectory optimization problem

While proceeding to the details in this section, the distinction between the terms *approaches*, *methods* and *techniques* will be provided in order to avoid misunderstanding. Each of these terms include a specific bunch of ideas which are illustrated in the map of trajectory optimization approaches as in Fig. 1.6.

In general, two types of approaches exist: *analytical approaches* and *numerical approaches*. Analytical approaches for the optimal trajectory result in analytical solutions. They can only be obtained in special cases, for example for very low-thrust orbit raising [69], and sometimes in the presence of some perturbations [38]. Therefore, results from the analytical approach are seldom feasible for most of the spacecraft trajectory optimization problems [4].

The majority of researches are dedicated to numerical approaches for spacecraft trajectory optimization problems [4]. These approaches can be divided in two well-known methods, called direct and indirect methods [12], [70]. Through direct methods, the solution is found in an approximate way based on the concept of parameterization on state variables $\mathbf{x}(t)$ and control inputs $\mathbf{u}(t)$. The parameterization concept usually involves discretization which concerns the process of transcription of the problem for transferring continuous

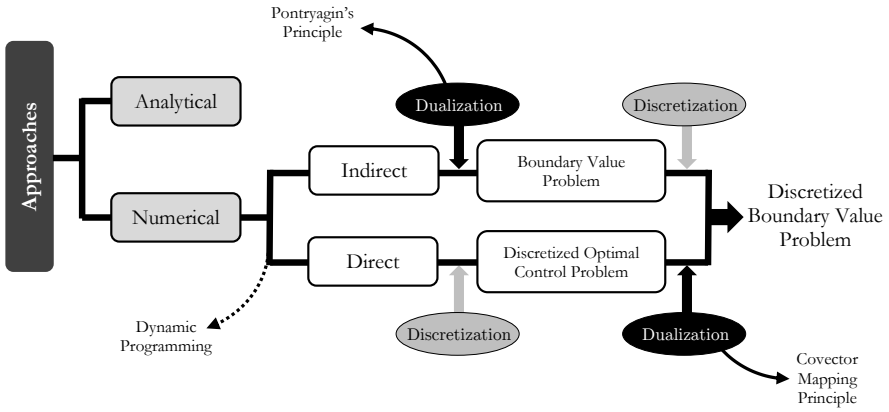

Fig. 1.6: Taxonomy of approaches in spacecraft trajectory optimization

functions, models, and equations into discrete counterparts. Certainly, such methods allow a candidate solution to be found, but no warranty is given about its optimality [12]. Indirect methods, on the other hand, use the same techniques and concepts as in the direct method, but have the feature of relying on necessary, analytic conditions for optimality. This allows the shifting of the optimization problem onto the determination of some parameters, known as the Lagrange multipliers, that should fulfill conditions for optimality at the beginning and at the end of the process. These variables are often defined as co-state, as they evolve along with the state vector. In other words, the main difference between direct and indirect methods is the involvement of co-state variables. The main issue related to indirect methods is the impossibility of knowing the initial guess, permitting the meeting of the boundary conditions. The search domain in such methods which is based on Pontryagin's Principle [4], is made even more complex by the fact that each Lagrange multiplier ranges in an unbounded set [12]. The connection between these two methods has been a challenge and a matter of interest within the community. The long lasting history regarding the bond of these methods has resulted in the recent advance in trajectory optimization [71]. This connection is described with respect to the *Covector Mapping Principle*, thanks to the efforts of Ross and Fahroo [72], [73], [74]. It describes the relationship between the multipliers of the discretized optimization problems and the co-states of the continuous optimal control problem.

Besides these two categories, sometimes dynamic programming is also considered as the third branch of numerical methods (dashed line in Fig. 1.6) in which the optimality criteria in continuous time is based on the Hamilton-Jacobi-Belman partial differential equation. However, most research in literature only consider direct and indirect methods as the only two branches of the numerical approaches [3], [4]. An overview of the approaches mentioned along

with their related methods, techniques and theories will be briefly discussed in this section.

1.4.1 Analytical approaches

Analytical approaches are the most desired ones since they usually provide solutions based on mathematical representations directly with zero approximation. However, they are not achievable most of the time due to the complexity of the problem. Such kinds of complexities may be because of the mathematical model or the objectives. The primary example of an analytical approach is the well-known Hohmann transfer for a simple orbit transfer mission. According to the proposed terminology and taxonomy in this review, Hohmann transfer is actually a very simple approach for transferring the spacecraft from one orbit to another for velocity increment minimization.

As for the analytical approach in a continuous domain, generally the process of achieving the optimal solution involves optimal control theory and relies on Pontryagin's Principle. The first step is establishing the problem in mathematical representation. This consists of determining the equations of motion, cost function, and applicable constraints. Constraints can be broken down into two primary types: path constraints and boundary constraints [75]. Path constraints represent restrictions on either the control or state at any time. For instance, engines have a finite amount of thrust yielding a maximum value for the control. It would be meaningless to look for a solution that requires a thrust more than the maximum available limit. Boundary constraints pertain to either the final or initial states. They may be given as a set of equality or inequality constraints. A state vector that does not violate any constraint is referred to as an admissible trajectory. Similarly, a control vector that does not violate any constraint is referred to as an admissible control. Then, the cost function is formed, augmented with Lagrange multipliers (or co-states) associated with the constraints and state differential equations of the system. Defining a convenient Hamiltonian, the first variation of the cost function due to differential changes in the control inputs is written. Next, co-state differential equations and boundary conditions are chosen to simplify this expression. This process of writing a problem in terms of the original variables and Lagrange multipliers (or states and co-states) is often referred to as dualization which makes the problem difficult to solve analytically. Such difficulties are well described by Ross [13]. However, this analytical approach is very useful as a sanity check for the numerical approaches.

As is shown in many references [12], [13], the number of Lagrange multipliers equals that of the state vector components. This means that even in the simplest case by considering the spacecraft as a material point, thus ignoring the attitude equations, the optimal set of Lagrange parameters could potentially be searched in $\mathcal{R}^7$ (six entries for the spatial coordinates plus one for the spacecraft mass). If even the initial epoch of the transfer is to be determined, the optimal set of parameters must be searched in $\mathcal{R}^8$.

Notable researches are dedicated to analytical approaches in spacecraft trajectory optimization. For example, Fernandez [76] developed a complete first-order analytical approach for the problem of optimal low-thrust limited-power transfers in an inverse-square force field between coplanar orbits with small eccentricities. The presented approach eliminates the singularity for circular orbits and can be applied for time-fixed transfers between coplanar orbits with small eccentricities.

1.4.2 Numerical approaches

By increasing the complexity of the model and the problem, the analytical approaches fade and numerical approaches become more favorable. A few of the popular methods in numerical approaches are discussed here, following the surveys [3] and [4].

1.4.2.1 Direct and indirect methods

The overall schema of numerical approaches is depicted in Fig. 1.7.

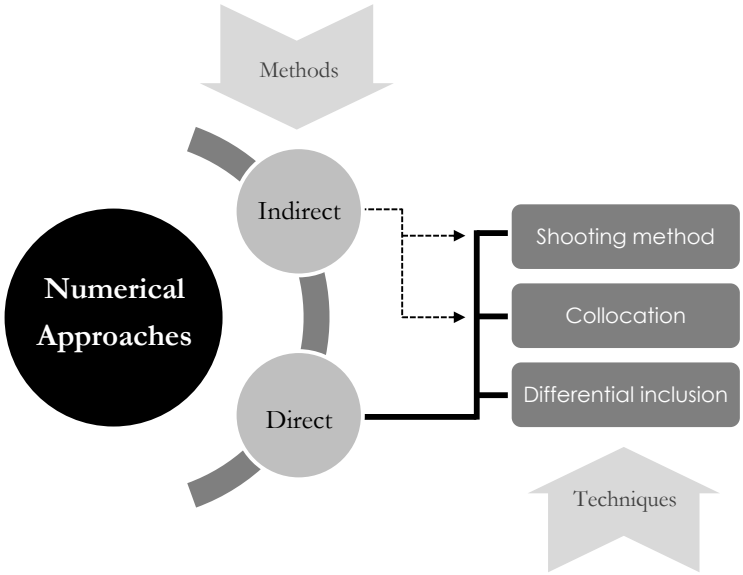

Fig. 1.7: Methods and techniques in numerical approaches

Two distinct branches of numerical approaches have arisen. Both branches attempt to minimize cost functions and constraint violations using discrete approximations [77]. This is performed by some gradient-based or metaheuristics

which will be discussed in the next section. The first branch is that of direct methods which transcribe the continuous optimal control problem into an optimization problem [11] considering states and control vectors. Satisfaction of the system equations is accomplished by integrating them stepwise using either implicit or explicit rules; in either case, the effect is to generate nonlinear constraint equations which must be satisfied by the parameters, which are the discrete representations of the state and control time histories. The problem is thus converted into a nonlinear programming problem [78]. Although direct methods are less accurate than indirect methods, the fact that they are easier to implement, have a larger domain of convergence, and have reduced problem size make them very attractive. One type of direct methods is the one when only the state variables are interpolated and control variables are considered in the objective function. Then a gradient-based technique or metaheuristic is employed to minimize the cost by changing the values of state variables. This method is sometimes referred to as the shape-based method, since it pertains to the shape of the state variables. Fourier series are very popular in this kind of method, specifically when applied to low-thrust trajectory optimization [30], [56], [55], [79], [80], [81]. The trajectory determined by the shape-based methods satisfies the equation of motion, boundary conditions, and even the constraint on the thrust acceleration. In addition, the solution of the shape-based methods can be shown to serve as a good initial guess for other approaches or methods. The resulting approach is very popular since it can be used in spacecraft trajectory optimization problems considering constraints or any other type of limitations. They have advantage as they can be easily manipulated for overcoming drawbacks of other methods. For instance, Xie et. al [31] used the shape-based approach in providing a new combination of the elevation-angle and radius shapes for the 3-D low-thrust trajectories using the initial orbital plane as the reference plane. This new shape combination avoids the two drawbacks of the spherical shaping method, including large out-of-plane motion and range control of state angle variation.

The second branch is that of indirect methods. An indirect method considers the dualized form of the equations including states and co-states within the time discretization. That is, the states and co-states are both considered within the discretization. While indirect methods typically enjoy greater accuracy than direct methods, three major problems arise. Firstly, the analytical forms of the necessary conditions must be expressed, including the co-state differential equations, the Hamiltonian, the optimality condition, and transversality conditions. Numerically speaking, this also makes the problem size large due to discretization of the co-states. Secondly, the analyst must guess certain aspects of the solution, such as the portions of the time domain containing constrained or unconstrained control arcs. Finally, this method also requires initial guesses for the co-states which decreases the domain of the convergence [11].

Previously mentioned, the very rarely considered branch is the dynamic programming method, which is seldom considered as a separate subdivision

of numerical approaches. The basic idea is to subdivide the problem to be solved in a number of stages. Each stage is associated with one subproblem, and the subproblems are linked together by a recurrence relation. The solution of the whole problem is thus obtained by solving the subproblems using recursive computations. For a more detailed insight in dynamic programming, the reader is referred to [82]. Dynamic programming has been extensively applied with success to discrete problems. Unfortunately, its application is severely restricted in the case of continuous state systems because of the *curse of dimensionality*; a term coined by Bellman to describe the problem caused by the exponential increase in the size of the state space [10]. Therefore, dynamic programming has not been successfully used in spacecraft trajectory problem with a large number of variables. Other methods, such as direct or indirect methods, must be employed. It should also be noted that regardless of whether a direct or indirect method is chosen, the states must be integrated from some boundary condition or the equations of motion must be enforced through constraints.

Both of the aforementioned direct and indirect methods aim at a high-fidelity solution, but may be time consuming for evaluating thousands of trajectories in the preliminary phase of the mission design. There have been various efforts and routes taken in overcoming the difficulties associated with the design of optimal spacecraft trajectories, for instance, by resorting to heuristics.

1.4.2.2 Numerical techniques

Hitherto, two different *methods* have been introduced within the category of numerical approaches. In this subsection, some numerical *techniques* will be briefly introduced, which can be used in direct and indirect methods.

One strictly direct technique is that of differential inclusions. The differential inclusions enforce the equations of motion at each discrete time by applying inequality constraints on the state derivatives [83]. These inequality constraints are obtained by substituting the upper and lower bounds on the control vector into the equations of motion. When the inequality constraints are met, the states at one node are said to lie in the attainable set at that node, given the state values at an adjacent node and the set of admissible controls. The advantage given by differential inclusions is that it effectively eliminates the explicit dependence on control values at each node. However, techniques such as this can become numerically unstable and the formulation can be problem dependent [3].

The shooting method is a well-known iterative technique to calculate the state histories given the control histories of the system. Most successful shooting applications have one salient feature in common, namely, the ability to describe the problem in terms of a relatively small number of optimization variables. One example is [48], where the direct shooting method is employed

with an evolutionary algorithm to solve the minimum-time orbit transfers of solar sail spacecraft for Mercury sample return missions.

Shooting methods can be divided into two types including shooting method and multiple-shooting method. The direct shooting method is a control parameterization method where the control is parameterized using a specified functional form, e.g.,

$$u(t) \approx \sum_{i=1}^{m} a_i \psi_i(t) \tag{1.46}$$

where $\psi_i(t)(i = 1, ..., m)$ are known functions and $a_i(i = 1, ..., m)$ are the parameters to be determined from the optimization. The dynamics are then satisfied by integrating the differential equations using a time-marching algorithm. Next, the cost function is determined using a quadrature approximation that is consistent with the numerical integrator used to solve the differential equations. The nonlinear programming problem that arises from direct shooting minimizes the cost subject to any path and interior-point constraints.

An extension of shooting the method is the multiple-shooting method. In a multiple-shooting method, the time interval $[t_0, t_f]$ is divided into $M + 1$ subintervals. The aforementioned direct shooting method is then used over each subinterval $[t_i, t_{i+1}]$ with the values of the state at the beginning of each subinterval and the unknown coefficients in the control parameterization being unknowns in the optimization. In order to enforce continuity, the following conditions are enforced at the interface of each subinterval:

$$x(t_i^-) = x(t_i^+) \tag{1.47}$$

These continuity conditions result in vector root-finding problem, where it is desired to drive the values of the difference between $x(t_i^-) - x(t_i^+)$ to zero. It can be seen that the direct multiple-shooting method increases the size of the optimization problem because the values of the state at the beginning of each subinterval are variables to optimize. This technique can also be applied in indirect approaches as well, where the co-states are also taken into account during discretization. The idea of the multiple-shooting method for both direct and indirect approaches is shown in Fig. 1.8.

Despite the increased size of the problem due to these extra variables, the direct multiple-shooting method is an improvement over the standard direct shooting method. The sensitivity to errors in the unknown initial conditions is reduced since the integration is performed over significantly smaller time intervals. Shooting methods are attractive because the equations of motion are enforced automatically by the marching integration. This effectively reduces the size of the problem by reducing the number of constraints that must be applied compared with collocation techniques which will be discussed in what follows [84].

Collocation techniques enforce the equations of motion through quadrature rules or interpolation. See [85] as a typical example. An interpolating function (interpolant) is solved such that it passes through the state values

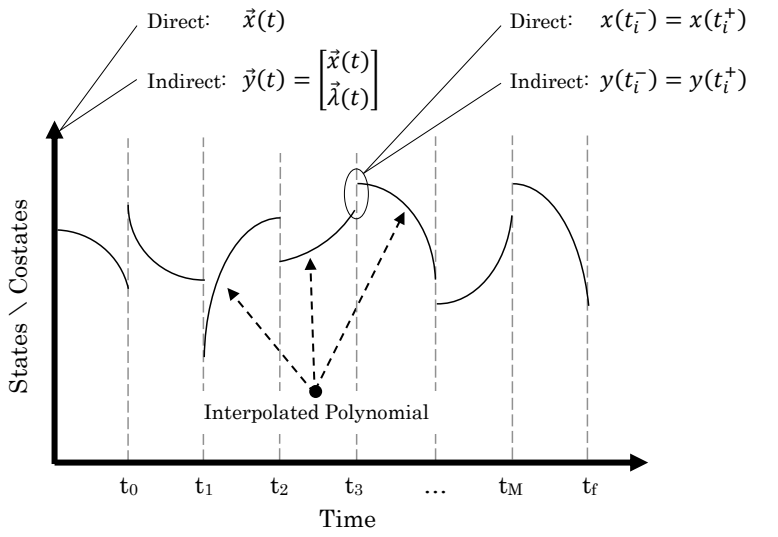

Fig. 1.8: Diagram of direct and indirect multiple-shooting methods

and maintains the state derivatives at the nodes spanning one interval (or subinterval) of time. The interpolant is then evaluated at points between the nodes, called collocation points. At each collocation point, an equality constraint is formed, equating the interpolant derivative to the state derivative function, thus ensuring that the equations of motion hold (approximately) true across the entire interval of time. This technique, sometimes referred to *transcription method* [11] is based on a trajectory discretization by small segments and on a near-uniform discrete approximation of thrust directions by a set of thrust profiles with an inequality constraint for each segment. The problem in this case can be stated as to minimize the total characteristic velocity with terminal conditions. The overall scheme of this process is depicted in Fig. 1.9.

In fact, the technique can be considered as a sequential nonlinear programming algorithm. This process has three fundamental steps. The first step is to convert the dynamic system into a problem with a finite set of variables. The second step is solving the finite-dimensional problem using a parameter optimization method (i.e., the nonlinear programming subproblem). Finally, the third step is to assess the accuracy of the finite-dimensional approximation and, if necessary, repeat the transcription and optimization steps. Actually this technique is a process including an *approach* (discretization), a *solution* (NLP) and a *minor technique* (refinement). Details about these steps are provided in [11].

In this technique, one fundamental step is to discretize the spacecraft trajectory as depicted in Fig. 1.10. According to Fig. 1.10, trajectory discretiza-

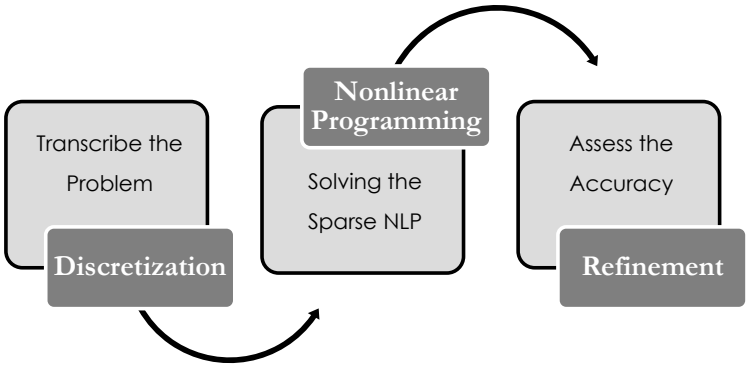

Fig. 1.9: Transcription technique process

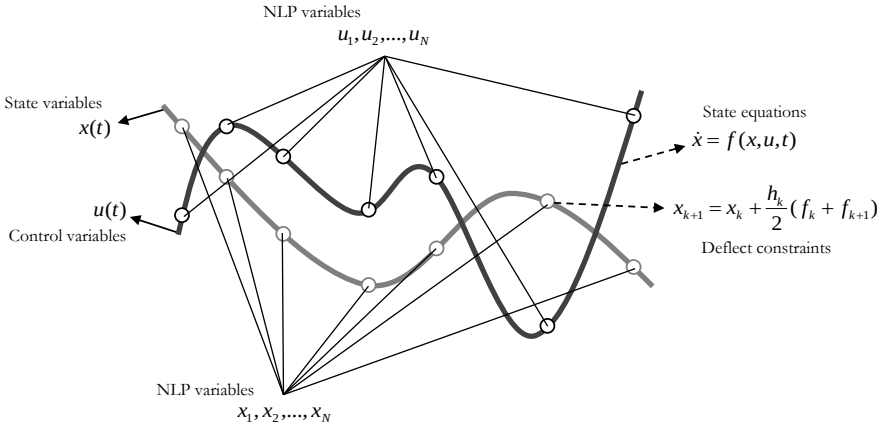

Fig. 1.10: Discrete versus continuous trajectory concepts

tion refers to a process which can be clearly seen as a multi-phase optimization problem. In this process, the trajectory is broken up into a finite number of legs and segments. The stage cost and constraints are generally expressed in terms of thrust magnitude and any violation from the maximum value. Transition functions can be obtained from the integration of the spaceflight equations of motion. The schematic representation of the corresponding trajectory structure is depicted in Fig. 1.11.

Once the states have been discretized and fitted with a polynomial, they are differentiated and then compared to the defined state derivatives at the collocation points. The difference is referred to as the defect. The defect is minimized in order to satisfy the specified equations of motion. This concept is the main notion in a class of techniques called *Pseudospectral methods*.

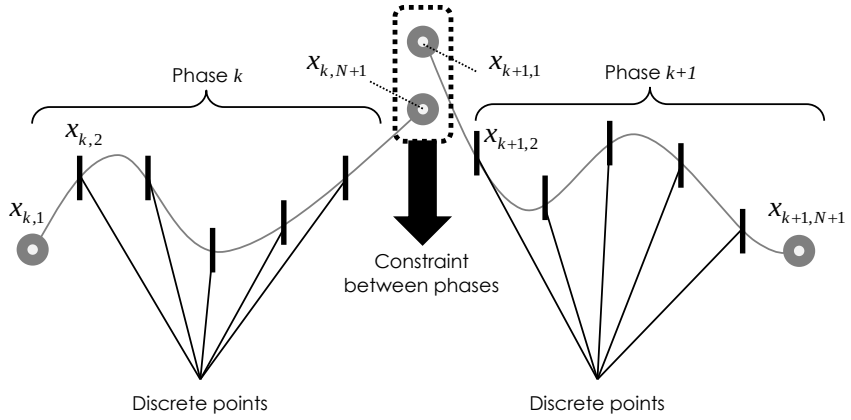

Fig. 1.11: Spacecraft trajectory discretization

Pseudospectral methods represent a class of direct methods that use collocation to solve optimal control problems numerically rather than analytically. This technique has become increasingly popular over the past several decades. In a pseudospectral method, the collocation points are chosen based on accurate quadrature rules and the basic functions are typically Chebyshev or Lagrange polynomials. They are generally known to converge spectrally as mentioned in [86]. In these methods, the discretization is accomplished by the use of global polynomials at collocation points. There are many sets that are commonly used in pseudospectral methods such as Legendre-Gauss (LG), Legendre-Gauss-Radau (LGR), Legendre-Gauss-Lobatto (LGL), Chebyshev-Gauss-Lobatto (CGL) and Hermite-Legendre-Gauss-Lobatto (HLGL). Although some researchers prefer the term *orthogonal collocation*, the terms *pseudospectral* and *orthogonal collocation* have the same meaning [87]. The reader is referred to [88, 89, 90, 91, 92, 93, 94, 95] and references therein for recent and comprehensive reviews of pseudospectal methods along with their applications in trajectory optimization problems.

Needless to say, each method may be more appropriate under certain conditions. The accuracy of such discretizations has been compared in [53]. However, the accuracies of individual methods are not rigorously examined in this survey since the main focus is not on the collocation method itself. While space precludes a more detailed discussion of collocation methods, the interested reader should consult the survey by Topputo and Zhang [96].

1.4.3 Summary

The approaches for spacecraft trajectory optimization problems are numerous and the current section is just a brief overview of current trends. As proposed in this section, when describing an approach for solving spacecraft trajectory

optimization problems, it can be either analytical or numerical. It is possible to employ analytical approaches for specific space missions by a simpler process than the procedure identified in optimal control theory. One process is semi-analytic, which is often employed in academic-strength problems. They usually use clever coordinate transformations and other *tricks* to avoid relying on Pontryagin's Principle or mitigate the complexity of optimal control theory. While such ad hoc techniques are indeed useful and important for the analysis of specific problems, they are not portable to the broader spacecraft trajectory optimization problems. When turning into numerical approaches, a method is often classified as either a direct method or an indirect method. As stated, the direct methods transform the spacecraft trajectory optimization problem to large-scale problems that require a high number of iterative computations. It constructs a sequence of points such that the objective function is minimized. In an indirect method, the process is also the same. However, it attempts to find a root of the necessary condition based on Pontryagin's Principle. It means it focuses on the adjoint variables in addition to state and control variables. Contrary to popular belief, Pontryagin's Principle itself is not an approach for achieving the solution. It is a problem generator which maps the optimal control problem to a boundary value problem by lifting it to a dual space [13].

It is also important to emphasize that there is no restriction with the method used to solve the problem and the techniques. For example, with the exception of the differential inclusion, one may consider applying a shooting or multiple-shooting technique to either an indirect or a direct method (notice the dashed lines in Fig. 1.7). The difference within the process for the shooting method is depicted in Fig. 1.12. Similarly, collocation techniques can be used not only in direct methods but also in indirect methods as well, with respect to the Covector Mapping Principle. The reader is urged to consult the works of Huntington et al. [53] for additional information. It should also be noted that the Covector Mapping Principle is satisfied by not only pseudospectral methods but also by Runge-Kutta methods. See [97] as an example in this regard.

When using an indirect approach, the user must compute the quantities of the Hamiltonian function. Unfortunately, this operation requires the user to have at least some knowledge of optimal control theory. Furthermore, even if the user is familiar with the requisite theoretical background, it may be very difficult to construct these expressions for complicated black box applications.

The major drawback for the indirect approach is the robustness. One difficulty is that the user must guess values for the adjoint variables, which is very non-intuitive since they are not physical quantities. Even with a reasonable guess for the adjoint variables, the numerical solution of the adjoint equations can be very ill-conditioned.

As for the direct method, more flexibility can be gained in finding the solution. Since the adjoint equations are not formed explicitly, analytic derivatives are not required. Instead, equivalent information can be computed using

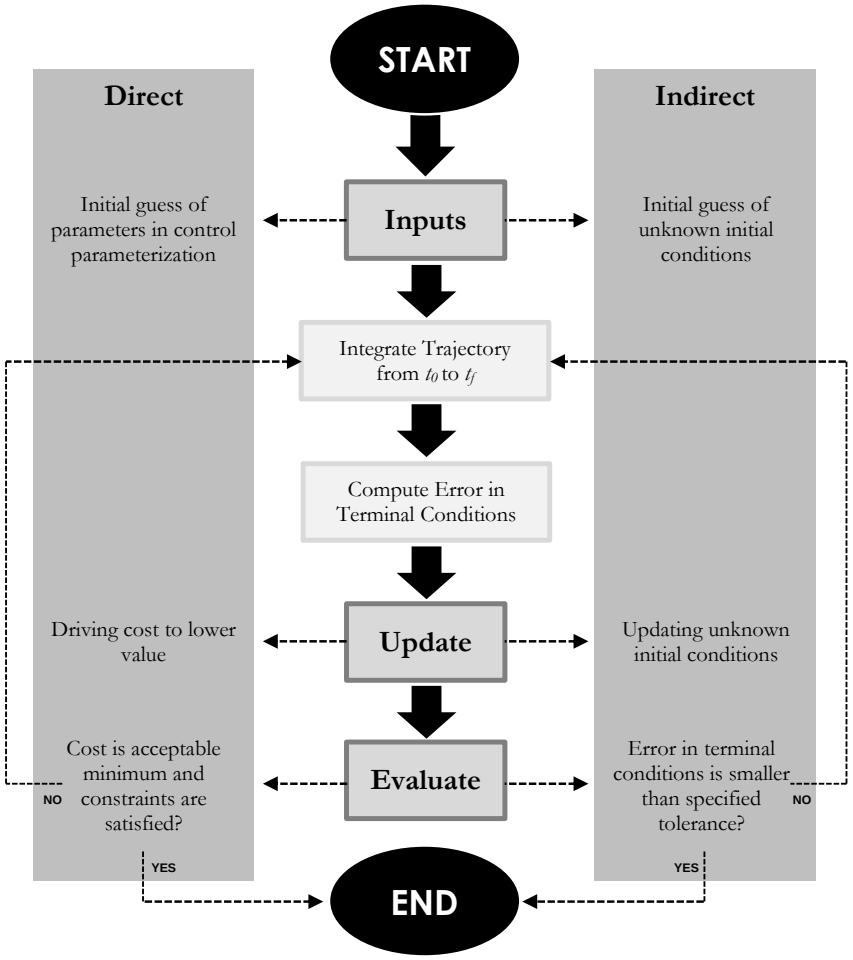

Fig. 1.12: Algorithm for direct and indirect shooting methods

sparse finite differences. Consequently, a user with minimal knowledge of optimal control theory can use the method. The method is flexible and new formulations are handled readily. Path inequalities do not require an a priori estimate of the constrained-arc sequence because the active set procedure automatically determines the arc sequence. The method is very robust since the user must guess only the problem variables. Furthermore, the globalization strategy, which is designed to improve a merit function, has a much larger region of convergence than finding a root of the gradient of the Lagrangian, which is the approach used by an indirect method. To sum up, for most applications, the direct method is quite powerful and eliminates the deficiencies

of an indirect approach. Indeed, it is often tempting to use a direct method simply because it is more easily implemented and understood. Nevertheless, there may be some situations that are problematic in direct strategies, such as singular arcs and discontinuous control.

The approach is the most important element of the four steps mentioned in the main process of solving a spacecraft trajectory optimization problem. The choice between the approaches vastly depends on what type of problem is being faced and what solution is actually expected. One good example which shows this dependency is the concept of on-line and off-line implementation of an approach [98]. The first kind of concept assigns maximum priority to the speed of convergence, maybe to the detriment of the accuracy or optimality itself. An example is provided by rendezvous with a moving target or docking maneuvers in which a real-time calculation of the approaching trajectory is often required [99]. As for the latter category, alternatively, the time for convergence is not as important as the optimality of the solution. An example is the long-time scheduling of a space mission involving multiple fly-bys [100].

1.5 Solution

The final step of the spacecraft trajectory optimization process consists of achieving the solution based on the approach developed in the previous section, either analytical or numerical. In Fig. 1.13, a hierarchy of algorithms used in the literature for spacecraft trajectory optimization is presented, which will be used as a scheme in the rest of the section.

If the analytical approach is developed in the previous step, the exact solution will be achieved. In the case of simple spacecraft trajectory optimization problems and sometimes very specific space missions, researchers usually use exact methods. Once again, the simplest example is solving the problem of impulsive orbit which arises with Hohmann transfer approach [17]. This analytical approach clearly ends in an exact solution directly. When a problem is solved analytically, it usually does not involve any iteration. There is no need to use any special numerical technique or iterative procedure to achieve the solution if the approach is truly analytical. As mentioned in the previous section, such cases are very rare in spacecraft trajectory optimization.

On the contrary, when spacecraft trajectory optimization problems become too complex for exact methods, numerical algorithms, heuristics and, in particular, metaheuristics are often used. In such cases, the problem is considered as a black-box which can be tackled with computational techniques and algorithms. Moreover, the problem imposes less restrictions to the application of different algorithms in this way.

1.5.1 Nonlinear Programming

Of the few types of computational techniques commonly used to solve trajectory optimization problems, gradient based methods such as nonlinear pro-

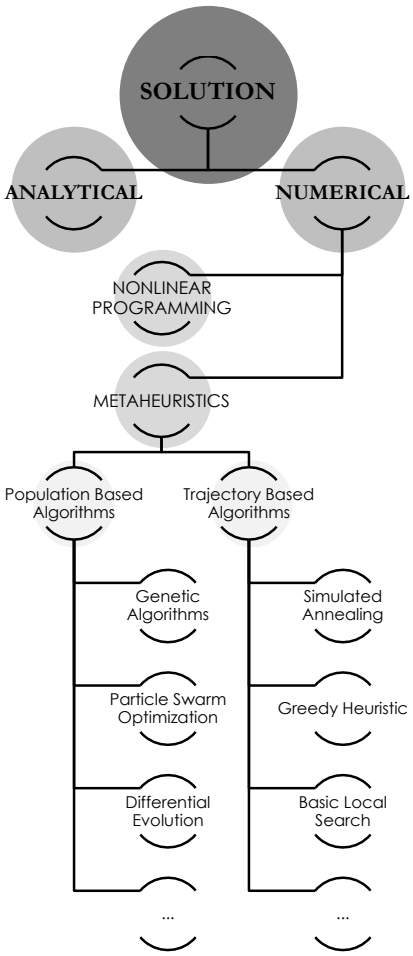

Fig. 1.13: Taxonomy of solutions in spacecraft trajectory optimization

gramming seems to be the most popular. Over the past decades, nonlinear programming (NLP) has become an indispensable tool for the optimization of space trajectories. These tools are now applied at research and process development stages, in the design stage, and in the online operation of these processes. More recently, the scope of these applications is being extended to cover more challenging, large-scale tasks. The ability to solve large-scale problems cheaply, even online, is aided by recent breakthroughs in NLP. They include the development of modern barrier methods, deeper understanding of line search and trust region strategies to aid global convergence, efficient exploitation of second derivatives in algorithmic development, and the availability of recently developed and widely used NLP codes.

Since NLP uses gradient information, it is often capable of relatively quick convergence and very accurate results (within the accuracy of the discretization). They also have well-defined convergence criteria. This has led to their popularity and the development of many individual software packages. A vast number of NLP solvers has been developed since 1970. Some of them are SNOPT [101], DONLP2 [102], filterSQP [103], rSQP++ [104] and KNITRO [105] which has been developed since the late 1990s until the mid 2000s. Another available package is the function *fmincon()* of MATLAB optimization toolbox. Developed in 2009, it is a general, multi-purpose constrained parameter optimizer for small, medium, and large spacecraft trajectory optimization problems [56].

The existence of various NLP solvers has led to a number of implementations and very reliable and efficient software packages for large-scale nonlinear programming. Certainly, one of the most recent and useful platforms for NLP is that of the NEOS Server [106]. This server provides state-of-the-art software free of charge to solve optimization problems. Other platforms such as AIMMS [107], AMPL [108], and GAMS [109] have also made the formulation and solution of optimization accessible to a much wider user base.

These NLP algorithms and associated solvers discussed in this section comprise only a sampling of representative codes, based on Newton-type methods. A complete listing is beyond the scope of this survey and the reader is referred to the NEOS software guide [106] for a more complete selection and description of NLP codes. Moreover, important issues such as scaling and numerical implementations to improve precision have not been covered here. Readers are referred to [110] for more information on NLPs.

The most noticeable problem with gradient based methods is that they require an initial guess of all the parameters of the system. Since all nodal state and control values are parameterized, the analyst must have some *a priori* knowledge of the optimal trajectory. The consequences of poor initial guessing are usually failure to converge or convergence upon a non-global optimum solution in the cost function.

Having the initial guess for the gradient-based method is a challenging issue in spacecraft trajectory optimization. A method to address this issue was first proposed in 1995 [111], which leads to the presentation of metaheuristics, an alternative way to solve the discrete direct or indirect trajectory optimization formulations.

1.5.2 Metaheuristics

In recent years, there have been significant advances in the use of metaheuristics to approximate solutions of spacecraft trajectory optimization problems. A metaheuristic is an iterative master process that guides and modifies the operations of subordinate heuristics to efficiently produce high-quality solutions [112]. It may manipulate a complete (or incomplete) single settlement or a collection of settlements in every iteration. The subordinate heuristics

may be high (or low) level procedures, or a simple local search, or just a construction method. The family of metaheuristics includes, but is not limited to, genetic algorithms, particle swarm optimization, simulated annealing, and their hybridizations.

Metaheuristics provide decision-making managers with robust tools that obtain high-quality solutions, in reasonable time horizons, to important applications specifically in aerospace optimization problems. The well-known survey by Blum and Roli [113] divides metaheuristics into two main categories including single solution algorithms and population based algorithms. The first category, also referred to trajectory-based algorithms, gathers local search, greedy heuristic, simulated annealing, tabu search, iterated local search, etc [112]. The second category, which is more practical in spacecraft trajectory optimization, regroups evolutionary algorithms such as genetic algorithms [114], evolution strategies [115], genetic programming [116], particle swarm optimization [117], etc. Survey by Xiong [118] also confirms this taxonomy with a slight difference in which the single solution algorithms are separated into two categories called trajectory based and multi-trajectory based algorithms. Other taxonomies exist as well that try to put algorithms in different groups [119], [120].

Among metaheuristics, evolutionary algorithms are particularly suited for most of the spacecraft trajectory optimization problems. A Generic diagram for most of the algorithms in this group is presented in Fig. 1.14.

The most recurrent instances in EAs group are Genetic Algorithms (GAs) which have been used in many papers [121]. Besides GAs, Differential Evolution (DE), has also been used frequently in the solution of spacecraft trajectory optimization. In these algorithms, the information about the structure of the problem can be incorporated in order to improve the efficiency of the algorithms.

The performance of different algorithms on different benchmark problems are compared in some researches. The tested algorithms in these studies include, besides GA and DE, EAs such as Particle Swarm Optimization (PSO), Ant colony optimization (ACO) and also local search algorithms such as Simulated Annealing (SA).

It can be highlighted that GAs are the first choice for most of the spacecraft trajectory optimization problems, perhaps due to their availability and ease of use. However, it is difficult to know the particular variant used (codification, operators, etc.) from the contents of the papers. Another observation regarding the literature confirms the fact that metaheuristics are more used in problems based on impulsive models rather than continuous models. To be more specific, gravity assist missions are the problems which are tackled mostly by evolutionary algorithms.

According to the literature, researchers tend to use EAs more than trajectory-based algorithms. However, the EAs have drawbacks which, to some authors, make it unacceptable as a primary means of trajectory optimization [3]. Firstly, as the algorithm is probabilistic, the difference in the output so-

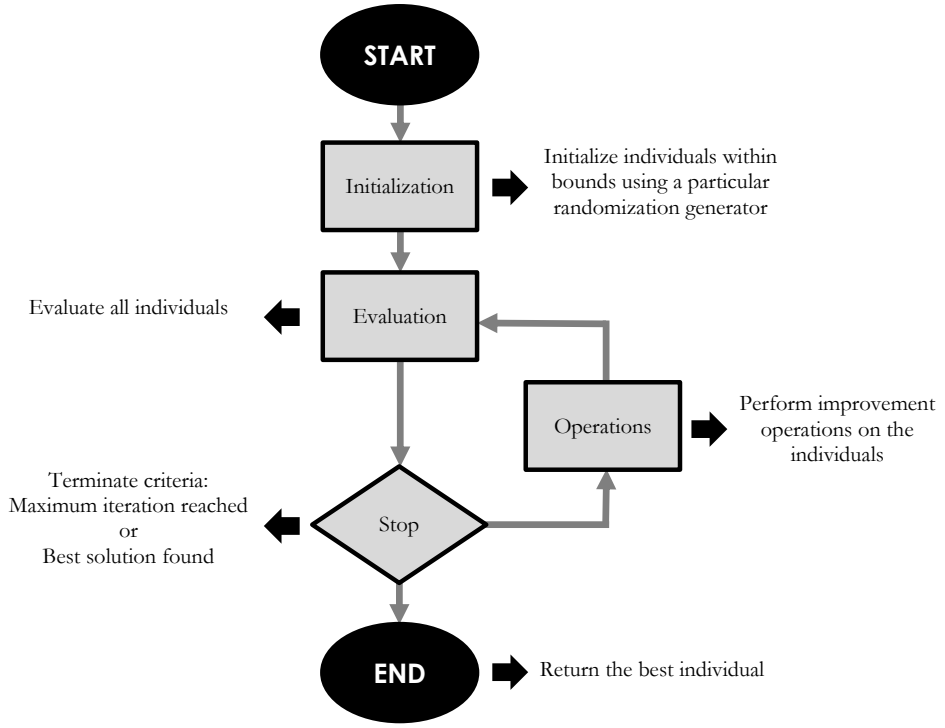

Fig. 1.14: Generic flowchart of evolutionary algorithms

lution in different runs can be high. Second, there are no well-defined convergence criteria (i.e., no necessary and sufficient conditions), such as those used in NLP algorithms.

It should also be highlighted that metaheuristics has been used not only for spacecraft trajectory optimization, but also for space orbit design problems as well. A typical example is the use of EAs for minimizing the average revisit time of a space mission over a particular target site during the specified days [122].

1.5.3 Hybrid algorithms

In recent years, many works have relied on cooperative (or hybrid) optimization techniques. In many cases, the best results are obtained with this kind of techniques, especially in real-life problems. At the beginning, cooperation was mainly carried out between several metaheuristics. But nowadays, more and more cooperation schemes between metaheuristics and exact approaches are proposed. These strategies usually give good results because they are able to

exploit simultaneously the advantages of several types of algorithms or methods. For example, it may allow giving quality guarantees to the identified solutions or may reduce the sensitivity to the initial guess [123].

The first attempt, in 1995 [111], was intended to be a solution of the multi-point boundary value problem (MPBVP) for systems with Mayer cost functions, and linearly appearing controls. The final numerical solution of the optimality conditions was initialized using guesses for the switching times, generated by examining the results of an NLP shooting technique initialized, in turn, by a shooting technique incorporated with GA. The solved problem was the reorientation of an inertially symmetric spacecraft. Other authors, such as those of [124], have opted to concentrate on modifying the components of the GA to improve the performance. Though the goal of that work was to improve the GA components themselves, the framework is relevant because it allows a more general representation of the control history in a direct shooting technique with a GA while using the result to initialize the NLP solution of a direct collocation method.

In the literature, for example [125] and [62], it is also demonstrated that a hybridization between global optimization techniques, generally applicable to black-box problems, with ad hoc branch-and-prune methods and exploiting the properties of the problem (e.g., continuity and differentiability, periodicity, symmetry, modularity) can greatly improve convergence, specifically when applied to spacecraft trajectory design. Hybridization may also refer to switching between global and local minimizers in an optimization algorithm in some researches [62], [126], [49]. However, it does not only refer to the algorithms. It sometimes refers to the approaches. For example, taking advantage of both direct and indirect approach features in solving a trajectory optimization problem in a space mission is sometimes called a hybrid technique [67], [33].

Analyses regarding the effectiveness of the hybrid algorithms are encouraging as, for the same computational effort (measured in number of function evaluations), hybrid algorithms were converging more accurately than common algorithms in many cases.

1.6 Summary and discussion

This review tried to provide a complete taxonomy of concepts within spacecraft trajectory optimization. This section provides the overall discussion regarding spacecraft trajectory optimization based on the literature referred to in this review along with some issues in this field. Moreover, future trends and useful considerations regarding the upcoming ideas in spacecraft trajectory optimization are put forward.

1.6.1 Highlights

By having an overview of the literature, several points can be highlighted. They include the relation between direct and indirect methods, metaheuristics and computational issues in spacecraft trajectory optimization which will be discussed briefly in this subsection.

The two branches of direct and indirect methods play an important role in categorizing the approaches. These two different philosophies have led to a dichotomy in the space community. Researchers who focus on indirect methods are principally interested in differential equation theory, while researchers who focus on direct methods are more interested in optimization techniques. While seemingly unrelated, there are a lot of common characteristics between these two methods. Specifically, as discussed in this review, in recent years researchers have delved quite deeply into the connections between the direct and indirect methods. They have discovered that the optimality conditions from many direct methods have a well-defined meaningful relationship. Thus, these two classes of methods are merging as time goes by. Covector Mapping Principle is actually the connectivity that fills the gap between these two branches [13]. It renders the terms direct and indirect obsolete in the modern viewpoint of spacecraft trajectory optimization and reveals that the obvious and shorter path of dualization first and computing afterward is strewn with difficulties while a longer path of reversing the operations eliminates the curse of sensitivity.

Regarding metaheursitics, the key point is selecting an algorithm intelligently for spacecraft trajectory optimization. The question remains unanswered about which algorithm is appropriate for a specific class of trajectory optimization problems. One reason is the definition of a *good algorithm*, which is different from paper to paper. Ideally, a *good algorithm* is capable of solving *most* of the instances of a given spacecraft trajectory optimization problem *faster* than a *bad algorithm* regardless of the method (direct or indirect) in an acceptable range of accuracy. Typically computer time is used to measure algorithm speed; however, when this is done it is imperative that all testing be done using the same hardware, compiler options, and operating system. The number of function evaluations can be used instead of (or in addition to) computer time, but then one must carefully define a *function evaluation*. Furthermore, in order to make a fair comparison between algorithms, it is important to consider several factors. These factors include testing a large suite of instances of the problem, using the same initial guess, and comparing them based on the same convergence criteria. When comparing one metaheuristic to another in a spacecraft trajectory optimization problem, it is common to perform benchmark tests using a suite of standard problems.

Four important computational issues that arise in the numerical approaches in spacecraft trajectory optimization are (i) consistent approximations for the solution of differential equations, (ii) the scaling of optimal control problem (iii), exploitation of sparsity in the problem, and (iv) computation

of derivatives of the objective and constraint functions. The manner in which the differential equations are discretized is of great important because an inconsistent approximation to the differential equations can lead to either nonconvergence or convergence of the optimization algorithm to the infeasible solution. Scaling and exploitation of sparsity in the problem are issues that greatly affect both computational efficiency and convergence of the algorithm. Finally, the manner in which derivatives are computed is also of great importance because accurate derivatives can greatly improve both computational efficiency and reliability.

1.6.2 Issues

During the last few decades, every year several papers regarding an innovative concept, approach or method for spacecraft trajectory optimization is published. The majority of the publications present truly novel ideas in this field. However, sometimes re-iteration of existing knowledge in this subject is introduced as a novelty. One example is the application of metaheuristics, more specifically EAs, in spacecraft trajectory optimization. Although some novelties are perfectly introduced in many publications, some papers are dedicated to the application of existing algorithms or techniques in specific space missions. The other similar issue is the parameter tuning of algorithms, which is called a novelty sometimes. Unfortunately, some of them do not represent a real advance in the field.

Another suggestion for improving the quality of the literature in this field is standardization of the whole evaluation process of the algorithms. It means using a high number of instances for evaluation of the proposed trajectory optimization algorithms or methods in articles. When performing empirical experiments with methods in spacecraft trajectory optimization, the goal is to show that a specific method performs better than other methods on a class of space mission instances with respect to some predefined objective, which is computational time and\or solution quality. However, an overview of the published papers in this field confirms that experiments are performed only on a few instances or even on a single instance, instead of using a class of space mission instances or randomly sampled orbits.

The next issue is the comparison of the results with other methods. The best way to show that a new method is really successful is to demonstrate that it outperforms state-of-the-art approaches. In some of the publications cited in this review, it is not clear whether a state-of-the-art method has been used for the comparisons or not. An alternative to show that the new method is at least competitive or interesting is to demonstrate that it outperforms some standard approaches. Unfortunately in most of the publications the proposed method is only compared to a few other approaches. Even more dramatic is the fact that in some papers no comparisons are performed at all.

There are some other issues as well, which are not discussed in detail here. For example, the experimental conditions are not completely clear in most of

the publications, sometimes best solutions are used for comparisons instead of average solutions, and statistical tests are not performed in most of the publications.

1.6.3 Suggestions for future trends

Several research lines can be identified as relevant subjects for future works which take the field forward in promising directions. This section is not by any means meant as a catalogue of or a roadmap for excellent research in spacecraft trajectory optimization, its only purpose is to point out some properties that the authors consider to be good research suggestions and practices, and some promising areas in which a lot of research is still needed.

As the first suggestion, any research on spacecraft trajectory optimization should be adequately framed in the general literature. Adequately framing a method entails deconstructing it, showing which components it consists of, and how these components were adapted to the specific space mission that is being solved. For this to be at all possible, it is promising to explain new spacecraft trajectory optimization approaches using the general optimization terminology as introduced in this review. Clarifying the four key items including model, objective, approach and the procedure for achieving the solution can help the readers understand the structure of the problem. This leads to deconstruction of the problem and reproducibility.

Following the proper terminology, all trajectory design should return to a situation in which methods are developed based on insight into the structure of the problem. Especially, research in spacecraft trajectory optimization should be applauded if it yields insight into the reasons why specific algorithms, methods or techniques work well on specific space missions. For example, the application of global optimization algorithms such as GA, PSO, etc. to space trajectory problems often considers the problem as a black box with limited exploitation of problem characteristics. However, in the component-based view of spacecraft trajectory optimization, concepts from one or a set of different frameworks can be combined into ever more powerful approaches and algorithms, such as hybrid approaches and algorithms, which are discussed previously. Such concepts can exploit problem characteristics, providing sensible improvements over the direct application of general purpose methods. The result of such a process is a deep insight into which components are responsible for the core optimization power of the overall method. Potentially, such analyses allow the spacecraft trajectory designer to draw important conclusions on why the method works as well as it does, by proving a relationship between the properties of the optimization method and the structure of the space mission problem that is being solved. Self-adaptive methods, specifically metaheuristics that automatically tune their parameters according to the dynamic stiffness of spacecraft trajectory optimization problem, also present an interesting line of future research.

Rather than spacecraft trajectory optimization, space mission design is also a challenging issue which is totally connected with spacecraft trajectory optimization itself. This process, better known as space mission planning, refers to two sequential steps; design the space mission and then optimizing the trajectory related to the mission. Many space mission planning problems are constructed in such a way that they include both real-valued variables and categorical variables. The categorical variables will typically specify the sequence of events that qualitatively describe the trajectory or mission, and the real-valued variables will represent the launch date, flight times between planets, magnitudes and directions of rocket burns, flyby altitudes, etc., [127]. For example in multi gravity assist maneuvers, the designer must choose both a set of discrete categorical variables defining the sequence of gravity assists, or flybys, to be performed and a set of real-valued variables that define the trajectory corresponding to that sequence of flybys. The categorical variables represent the planets chosen for gravity assists, and the real-valued variables represent other parameters relevant to the trajectory, such as the date of launch and the flight times between each planet in the sequence. Therefore, the mission planning can thus be considered as two nested optimization problems including an inner loop that optimizes the trajectory for a given mission sequence, and an outer-loop scheduling problem that chooses the optimal sequence of gravity assists. Such concepts involve using EAs in both combinatorial and continuous domain, which is an interesting topic for future researches, since little research is dedicated to them.

Another interesting subject is that of scalarization techniques in spacecraft trajectory optimization, which has not received much attention in the literature. Multi-objective trajectory optimization methods utilize various scalarization functions in different researches depending on the space mission, approach, type and number of minor cost functions. In most scalarization functions, preference information of the decision maker is taken into consideration. After the scalarization phase, the widely developed theory and methods of single objective optimization can be used to deal with the problem. However, no current research can be found regarding the comparison of different scalarization techniques in spacecraft trajectory optimization, or at least in a specific space mission.

The other new trend is designing global optimization metaheuristics which are useful in automatically finding and selecting good trajectory options between the many often possibilities one has in the preliminary phases of mission design. Their use and efficiency are established for chemical propulsion problems of high complexity (i.e., large launch windows and multiple flybys) whenever approaches more sophisticated than the straightforward use of standard algorithms are adopted. Preliminary results in this sense are already available and point to an increased need for computational resources. It seems likely that future research results will aim at proving the use of these techniques for the automated computation of low-thrust trajectories as well.

Comparing different trajectory optimization approaches has so far been a largely unstructured affair, with testing procedures being determined on the fly and sometimes with a specific outcome in mind. Although some authors have developed procedures to make a statistically sound comparison, widespread acceptance of such procedures is lacking. Perhaps a set of tools is needed, i.e., a collection of optimized trajectories or libraries specifically designed to determine the relative quality of a set of methods on a set of problem instances. These should both be easy to use, and their results should be easy to interpret. Until such tools are available and a specific comparison protocol is enforced by journal editors and reviewers, the door is left open for researchers to select the method of comparison that proves the point it is intended to prove. Moreover, such contributions can be published even if they do not contain any novel method or a method that outperforms all existing approaches.

1.7 Conclusion

A review for solving spacecraft trajectory optimization problems has been given in this chapter. The solving process is decomposed into four key steps of mathematical modeling of the problem, defining the objective functions, development of an approach and obtaining the solution of the problem. Using these steps, several subcategories for each step have been described. Subsequently, important classifications and their characteristics have been discussed for solving spacecraft trajectory optimization problems. Finally, a brief discussion has been given on how to decide and choose in each step.

This review is considered complementary to most of the previously published survey articles on spacecraft trajectory optimization. It reflects most of the research and efforts that has been carried out over the past decade while simultaneously providing a summary of the vast amount of work that was done up to this point. The material in this review has been presented to give the reader an understanding of how methods, techniques and algorithms are categorized for spacecraft trajectory optimization problems. It is also worth noting that a great deal of discussion has been given to the distinction between different categories, not just in this review but also in previously published surveys.

To sum up, trajectory design and optimization has a broad variety of applications in different fields, particularly in aerospace engineering. The solution of a trajectory optimization problem that minimizes a cost function subject to nonlinear differential equations of motion and various types of constraints may be obtained by either an analytical approach or a numerical approach. From the viewpoint of numerical computation, spacecraft trajectory optimization is a hard global optimization problem, which is even more difficult when the analytical expressions of the objective function or the constraints are not usually available. Moreover, even simple bi-dimensional cases display an enormous

number of local optima. All of the iterative techniques and algorithms for spacecraft trajectory optimization can present convergence difficulties (non-convergence, slow convergence, etc.). These difficulties should be considered for each specific problem since a general technique for all spacecraft trajectory optimization problems does not exist. It does not make sense to ask general questions such as "Are direct methods better than indirect methods? "or "Is Genetic Algorithm better than Particle Swarm Optimization in spacecraft trajectory design?". The answer to such questions can only be "It depends on the space mission and the mission requirements". This is not to say that all approaches, methods and algorithms are equally powerful, nor that it is impossible to obtain meaningful insight into whether a specific method is more suitable for solving a specific class of trajectory optimization problems than another. Viewing spacecraft trajectory optimization concepts as sets of general ideas allows a broader view of the literature and allows for the discovery of similarities between the structure and inner workings of methods that remain opaque if only the label the author of the method has chosen for it is considered. This is certainly true in the modern view of spacecraft trajectory optimization, in which ideas may combine concepts from different frameworks and the framework that is used to name the method is a matter of the author's personal opinion. Choosing a method for solving the spacecraft trajectory optimization problem is based largely on the type of problem to be solved and the amount of time that can be invested in coding. Various extensions to the currently employed approaches offer opportunities and challenges for future works.

2

Long-range Space Rendezvous

2.1 Introduction

2.1.1 Motivation of Research

Rendezvous orbital dynamics is a key operational element for complicated space missions, which has a research history of several decades, and many novel research ideas and results on this topic are still appearing in different missions such as asteroids explorations [128], Mars missions [129] and Earth orbit transfers [130]. The general long-range rendezvous problems are usually solved based on the Lambert method which has been one of the most extensively studied methods for decades and still is a subject of interest in many researches [131, 132]. In a long-range rendezvous, the spacecraft is expected to have an orbital maneuver where, in the general case, all of the orbital elements involved suffer changes. This type of non-coplanar orbit transfer problem is the early phase of the overall space rendezvous [6]. In this type of mission, a two-impulse transfer obtained by the Lambert method that starts on the initial orbit and ends on the final orbit within a specific time can be the fuel-optimal transfer. However, for some specific cases, such as bi-elliptic Hohmann transfers, it is analytically possible to have the same or less fuel consumption than a two-impulse transfer for specific missions [15]. In either cases, no observation is applied to the magnitude of impulses, and thus, the solutions might not be feasible in scenarios where impulse limits are considered. In fact, the problem becomes more challenging when propulsion systems with low impulses are used in such non-coplanar transfers. When the goal is to find the optimal transfer in terms of fuel and time complying a given impulse limit, finding the global optimal solution becomes challenging [133]. One effective option to deal with this difficulty is to use meta-heuristics and evolutionary algorithms [134, 135]. As a result, a more thorough investigation is needed to find the best solution using these numerical methods and techniques [2].

2.1.2 Recent Advances

In recent years, several attempts have been made to deal with impulsive transfers in various space missions and conditions with different techniques [136, 137]. Such conditions or constraints can be the actuation uncertainties [138], collision avoidance constraint [139, 140] or other criteria. According to the literature, this multi-objective problem generally includes two types of costs and two types of constraints. The costs are fuel and time while the initial and final orbits along with allowable impulse limit are considered as constraints. Besides the impulse magnitude constraint, impulse direction is also sometimes considered as another kind of constraint in the literature. It can be the tangential impulse [141], along-track impulse [142] or continuous-thrust transfer with specific control magnitude and direction constraints [143]. Regarding the initial and final condition, the problems can be categorized as point to point, point to orbit and orbit to orbit maneuvers. A lot of research has been dedicated to such problems. However, an efficient approach in facing the problem is that which can be used regardless of the problem type. In [144], the problem of multi-impulse transfer is tackled by an analytical solution based on polynomials. However, the transfer time is not considered, and the approach is tested on special cases. Coplanar two-impulse rendezvous is studied in [145] and [146]. The research in [147] focused on a homotopic targeting technique for space rendezvous. Although it adequately considered the presence of orbital perturbations, the impulse limit is not taken into account. More research can also be found in the literature in which either the time, the impulse limit or other criteria have been taken into consideration besides the fuel consumption [148, 149].

2.1.3 Main Contribution of the Research

Impulsive rendezvous between non-coplanar orbits considering fuel, time and impulse limits makes the problem multi-objective and challenging to solve. To that end, a metaheuristic algorithm combined with an analytical solution of the multi-impulse transfer has been considered. The developed approach minimizes both the fuel and time, keeping the minimum necessary number of impulses without violating the impulse limit in a long-range rendezvous mission. Particularly, the proposed strategy in the current context is a direct approach based on the discretized Lambert problem and a novel hybrid self-adaptive evolutionary algorithm. In this approach, as the first step, the problem is solved, disregarding the impulse limit and transfer time and a solution that minimizes the total fuel consumption is achieved. This solution can be obtained either analytically or by means of an NLP or an evolutionary algorithm. Having this solution, the Lambert problem associated with the obtained transfer trajectory is extended to multi-impulse transfers in which the overall transfer is divided into a specific number of stages. Then, an analytical scheme is proposed based on dividing the velocity increments into the

necessary number of small impulses at the intersections of space orbits corresponding to each stage. This solution results in a specific sequence of impulses within the proposed approach, which is feasible in terms of fuel consumption. Considering this solution as a seed solution for generating individuals with high quality, a robust self-adaptive evolutionary algorithm is proposed. The algorithm benefits from several enhancements over the standard evolutionary techniques by hybridizing an NLP solver with a modified Particle Swarm Optimization. The algorithm is constructed as a self-adaptive technique since its parameters are auto tuned according to the orbital parameters of the initial and final orbits as well as the specified impulse limit for the space rendezvous mission. Combined with the proposed approach, the algorithm is tested on a wide set of long-range space rendezvous missions with various impulse limits. Results indicate that the algorithm is capable of decreasing the overall transfer time while it satisfies the impulse limit and holds the optimal fuel consumption.

The rest of the chapter is organized as follows. Section 2.2 introduces the discretized Lambert problem for multi-impulse orbit transfers. The simple feasible solution, obtained by an analytical approach is proposed in Section 2.3 based on dividing the velocity increments within the intersections of space orbits. In Section 2.4, the robust self-adaptive evolutionary algorithm, along with the improvements in its structure, are discussed. Section 2.5 provides the simulation results obtained by utilizing the proposed approach in several long-range space rendezvous missions. Finally, the conclusions are provided in Section 2.6.

2.2 Discretized Lambert Problem

2.2.1 The Approach

Consider a general long-range rendezvous with initial and final orbital elements as $p_0 = [a_0, e_0, i_0, \Omega_0, \omega_0, \nu_0]$ and $p_f = [a_f, e_f, i_f, \Omega_f, \omega_f, \nu_f]$. Since two orbits do not have any intersections and no orbital elements are the same between the two orbits, a two-impulse transfer trajectory that minimizes the total fuel consumption exists. The transfer trajectory starts from the initial true anomaly of θ_i of the initial orbit and ends in the final true anomaly of θ_f of the final orbit. These two anomalies correspond with two radius vectors $\mathbf{r}_i$ and $\mathbf{r}_f$. Besides these two anomalies, a third parameter fulfills a complete Lambert problem regarding any desired approach. The third parameter can be semimajor axis [150] (Lagrange transfer-time equation), eccentricity [151] (Avanzini's approach), a universal variable [152] (Izzo's approach), flight-path angle [153] or any other parameter that constructs a unique Lambert problem. Considering the Lagrange transfer-time equation, the unknown set of variables can be established as $x = [\theta_i \quad \theta_f \quad t]$ when only the orbit transfer is desired. If the actual rendezvous is also desired besides the orbit change, the unknown

variables will be $x = [\theta_i \quad t]$ with respect to the relative phase difference of chaser and target. In either case, the problem can be turned into a blackbox optimization problem with total velocity increment Δv as the output, which is the summation of two impulses at intersections $\Delta v = \Delta v_1 + \Delta v_2$.

By means of an effective nonlinear programming (NLP) method or an evolutionary algorithm (EA), the best solution with minimum fuel could be approximated. However, no penalty is considered for the excessive magnitude of Δv_i when low-impulse propulsion systems are utilized. In order to find the solution with minimum fuel and transfer time in the multi-impulse maneuver with respect to a given impulse limit, a new approach is proposed as shown in Fig. 2.1.

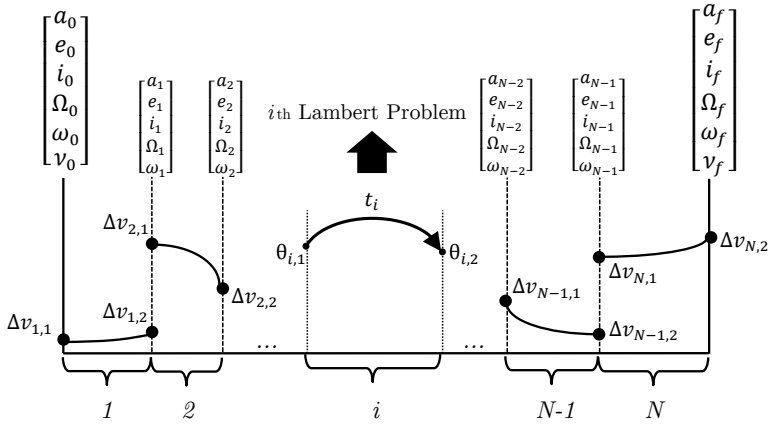

Fig. 2.1: Scheme of discretized Lambert problem.

In this approach, the original Lambert problem is divided into N number of stages. Each stage represents a unique Lambert problem with unknown Lambert problem variables. Having N number of stages will generate $N-1$ intermediate orbits, represented by a_i, e_i, i_i, Ω_i and ω_i ($i = 1$ to $N-1$), along with N jumps. Every jump is denoted by initial and final anomalies as $\theta_{i,1}$ and $\theta_{i,2}$ corresponding to the initial and final state vectors and the transfer time t_i in each stage. By considering the orbital elements of stages and the Lambert problem variables as the inputs, a complete multi-impulse Lambert problem with $2N$ impulses ($\Delta v_{i,1}$ and $\Delta v_{i,2}$) will be fulfilled. Therefore, the decision variables, denoted by the vector x, will be formed as:

$$x = x(a_i, e_i, i_i, \Omega_i, \omega_i, \theta_{i,j}, t_{i,j}) \tag{2.1}$$

According to this approach, a total of $2N-1$ sets of orbital elements will be known when a solution is achieved. These sets contain $N-1$ intermediate

orbits, which are the inputs of the problem, and N jumps, which are trajectories representing the solution from each stage that represents a minor Lambert problem. The $3N$ number of variables are associated with N Lambert problems, and $5(N-1)$ variables are associated with the shape and orientation of the $N-1$ intermediate orbits. Therefore, a total of $8N-5$ decision variables are associated with the optimization of multi-impulse transfer in this approach for N stages. Recalling that each stage has two impulses, the total number of variables for n number of impulses ($n=2N$) will be $4n-5$.

The initial and terminal conditions can be easily handled for various types of long-range rendezvous missions in this approach. This can be done by adjusting the initial true anomaly of the first stage ($\theta_{1,1}$) and the final true anomaly of the last stage ($\theta_{N,2}$). As for orbit to orbit rendezvous, the initial true anomalies of two spacecraft ν_0 and ν_f are unknown and therefore $\theta_{1,1}$ and $\theta_{N,2}$ are free. In this case, the problem will be tackled according to the described optimization variables. For orbit to point rendezvous, when it is necessary for the two spacecraft to be in the same true anomaly in the final orbit, the parameter ν_f is known, which is the initial position of the target spacecraft in the final orbit at epoch. In this case, $\theta_{N,2}$ will be fixed and its value can be calculated based on the total coast times and the initial true anomaly of the target spacecraft ($\theta_{N,2}=\theta_{N,2}(\nu_f,t_i,\bar{t}_i)$). Similarly, if it is required that the first impulse of the chaser occurs at a specific true anomaly in the initial orbit (ν_0 is known), the variable $\theta_{1,1}$ will be fixed. To sum up, depending on what sort of long-range rendezvous is the subject of the problem, the two optimization variables ($\theta_{1,1}$ and $\theta_{N,2}$) can be either fixed or free.

This approach has some advantages over the traditional methods. First, as with most of the multi-impulse approaches based on the Lambert problem [50], the total number of inputs is lower in comparison to the traditional approach in which the direction, magnitude and time of impulses are considered as the decision variables [154] (total of $4n$ decision variables). As an example, in the traditional approach for a two-impulse transfer, the time of acting for each impulse along with the impulse vector, including the magnitude and two angles representing the direction of the impulse in three dimensions are considered as the decision variables. However, in the current approach, only three variables are required including the initial true anomaly, the final true anomaly and the transfer time. Although the number of variables is lower, the approach comes with the burden of tackling the Lambert problem in achieving the solution, which requires iterations. This is due to the fact that, by defining multiple minor Lambert problems, the majority of the characteristics of the transfer trajectories wil be revealed. In other words, the shape of the transfer trajectories are taken into account instead of the impulse directions in the Cartesian coordinate system. Regarding this fact, this approach can be referred to as an impulsive shape-based approach.

The next advantage is handling the terminal conditions. The initial and final condition for point to point, point to orbit and orbit to orbit cases can be easily handled in the current approach by setting the Lambert problem

variables as free or fixed in the first and last stages. However, satisfying the terminal condition in the traditional approach is an issue which usually needs to be considered as an additional term in the objective function. Besides, since the shape of stages is defined via the actual orbital elements with physical meanings, the method benefits from rapid convergence as the orbital elements have known boundaries in real applications.

2.2.2 The Objectives

As the boundary conditions are already satisfied by the proposed approach, three types of objectives are defined for the problem including fuel, time and impulse violation. Regarding the proposed approach, the overall fuel consumption in terms of Δv in every stage is denoted by J_f and is defined as:

$$J_f = \sum_{i=1}^{N} \left(\Delta v_{i,1} + \Delta v_{i,2} \right) \tag{2.2}$$

The total transfer time is the summation of all coasting times between the impulses. Regarding the proposed approach, two types of coasting times exist, denoting by t_i and $\bar{t}_i$. The first type (t_i) is the time associated with each minor Lambert problem in every stage between $\Delta v_{i,1}$ and $\Delta v_{i,2}$. The latter ($\bar{t}_i$) is the time between the impulses $\Delta v_{i,2}$ and $\Delta v_{i+1,1}$, in which the spacecraft travels between two sequential Lambert problem. Therefore, the time objective, represented by J_t, is defined as:

$$J_t = \sum_{i=1}^{N} t_i + \sum_{i=1}^{N-1} \bar{t}_i \tag{2.3}$$

The impulse violation regarding a given impulse limit needs to be calculated for each Δv in every stage. As a result, the penalty denoted by the jth Δv in ith stage is calculated as

$$J_{i,j} = \frac{1 + sgn(\Delta v_{i,j} - \eta)}{2} (\Delta v_{i,j} - \eta) \tag{2.4}$$

where η is the given allowable impulse during the orbit transfer. In this equation, the typical *sign* function is used to extract the appropriate penalty for each impulse. According to this equation, if the impulse is less than the predefined limit ($\Delta v_{i,j} < \eta$), the penalty associated with that impulse becomes zero ($J_{i,j} = 0$). On the other hand, if the impulse exceeds the predefined limit ($\Delta v_{i,j} > \eta$), the associated penalty will be equal to the amount of exceeded impulse magnitude ($J_{i,j} = \Delta v_{i,j} - \eta$).

Consequently, the overall magnitude of the penalty function due to the impulse violations in all stages is calculated as:

$$J_v = \sum_{i=1}^{N} \left(J_{i,1} + J_{i,2} \right) \tag{2.5}$$

Having the cost functions, the overall objective function can be written via scalarizing the three objectives as:

$$J = J_f + \zeta J_v + \xi J_t \tag{2.6}$$

where ζ and ξ are scalarization coefficients for impulse violation and transfer time respectively. In the literature, the impact of the choice of the underlying scalarizing coefficients is still far from being well understood in space orbit design and optimization problems. Due to this matter, it is very important and crucial to choose these parameters according to the type of the space transfer. To demonstrate the effect of these weighting coefficients, one example of feasible and non-feasible solutions within the solution domain of a sample four-impulse rendezvous with one intermediate orbit is represented in Fig. 2.2 and Fig. 2.3. In this case, an orbit to orbit transfer with the initial and final orbits as $p_0 = [10000, 0.1, 30, 40, 55]$ and $p_f = [16000, 0.4, 25, 50, 30]$ is considered with impulse limit of $\eta = 0.5 km/s$.

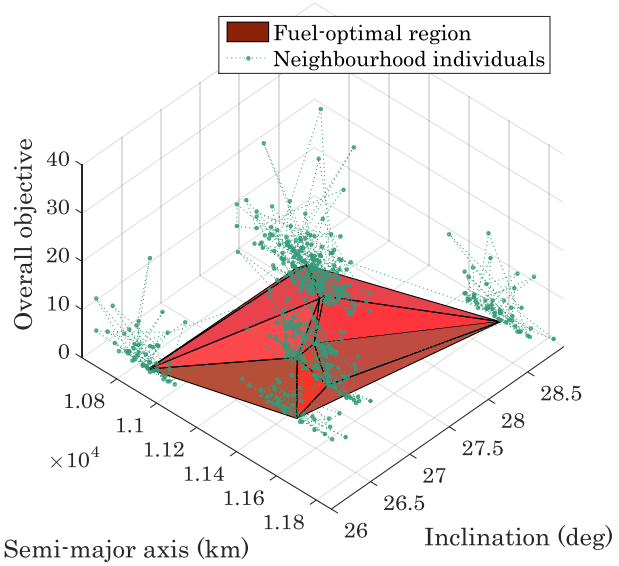

Fig. 2.2: Solution domain of J ($\xi = 0$) in a four-impulse rendezvous ($\Delta v = 1.4777 km/s$).

In Fig. 2.2 the solutions are plotted as a function of semi-major axis and inclination of the intermediate orbit. In this figure, the surface associated with a set of good solutions (obtained by means of the approach in Sec. II

and IV) is plotted, which includes the solutions with minimum fuel in this case. In addition, neighboring solutions of the surface are plotted as green points. Note that the time objective function has been disregarded ($\xi = 0$). The points correspond to the solutions with larger objective values either due to impulse violation or fuel consumption. Extracting the points near the minimum-fuel region with respect to a selective threshold and recalculating J and J_t for $\xi \neq 0$ are shown in Fig. 2.3. In this figure, the objective representing the total transfer times (J_t) for fuel-optimal region versus the overall cost is depicted.

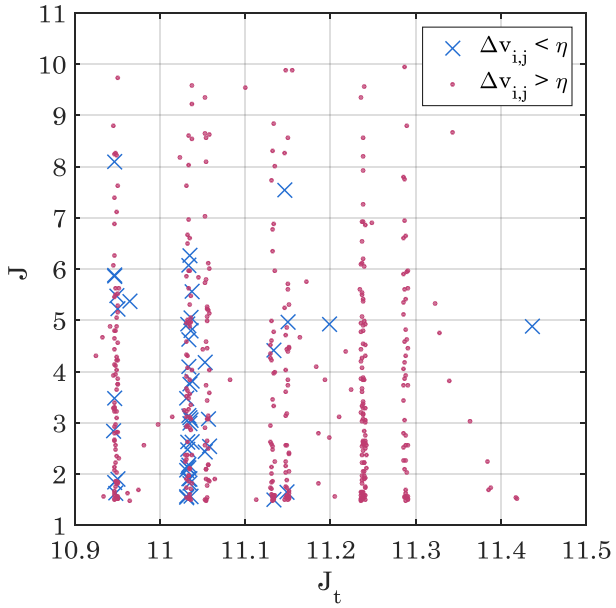

Fig. 2.3: Time-optimal solutions in the neighborhood of fuel-optimal region.

In this figure, the solutions for acceptable transfers satisfying the impulse limit are plotted along with non-feasible solutions in this matter. Local optima regions can be obviously distinguished in Fig. 2.3 showing that the consideration of ζ significantly affects the desired region and reaching the global optimal solution is challenging in this type of problem. It is worth noting that the best values of ζ are different from case to case. The major challenge is that the tuned value of this parameter along with ξ for one space rendezvous does not necessarily enhance the search process in another rendezvous problem. Therefore, these parameters have to be tuned automatically according to the initial and final orbits in each rendezvous mission. Such self-adaptive concepts makes

the approach robust to every unique Lambert problem with a given impulse limit.

2.3 Simple Feasible Solution

Regarding the proposed approach, the problem can be solved via an EA. However, two issues will arise. First, the solution domain of the optimization problem becomes too large when a high number of impulses are desired. The reason is that by considering very low-impulse transfers, the number of stages is increased which consequently increases the number of optimization variables. Following this, the landscape of the problem becomes chaotic, which makes the convergence process slow and the quality of the final solution can drop dramatically. The second issue is that the minimum number of impulses required for the entire transfer is unknown.

In order to deal with these issues, individuals near the solution which minimized the fuel consumption can be used to improve the quality of the initial populations of the EA and also to calculate the necessary number of impulses regarding the given impulse limit. Seeding the EA based on a feasible solution derived from the fuel-optimal region will effectively improve the convergence and the optimality of the algorithm. This is due to the fact that the newly generated populations have small values for one or two objectives while satisfying the impulse limit, forcing the algorithm to minimize the time near the fuel-optimal region. The seeding technique can be used either within the initial population at the beginning of the optimization, or during the optimization process when the diversity of the population is less than a predefined threshold, or both. To implement this concept, the problem is solved for N_0 number of stages without considering time ($\xi = 0$) and impulse violation ($\zeta = 0$) initially. This solution can be derived from either an existing analytical approach or a numerical solution. Having the solution for $N = N_0$, the $\Delta v_{i,j}$ at the intersection of transfer orbit with N_ith and N_{i+1}th intermediate orbits can be divided into a necessary number of minor impulses. A schematic view of this concept is depicted in Fig. 2.4.

Following this process, the impulse at each intersection is divided into minor ones keeping the impulse direction fixed. The required number of divisions can be calculated based on the given impulse limit η as:

$$\varphi_{i,j} = \lceil \frac{\Delta v_{i,j}}{\eta} \rceil \tag{2.7}$$

where $\varphi_{i,j}$ is the minimum number of required impulses. Accordingly, the actual impulse division at intersections for $1 < k < \varphi_{i,j}$ can be represented by:

$$\sum_{k=1}^{\varphi_{i,j}} \Delta v_{i,j,k} = \Delta v_{i,j} \tag{2.8}$$

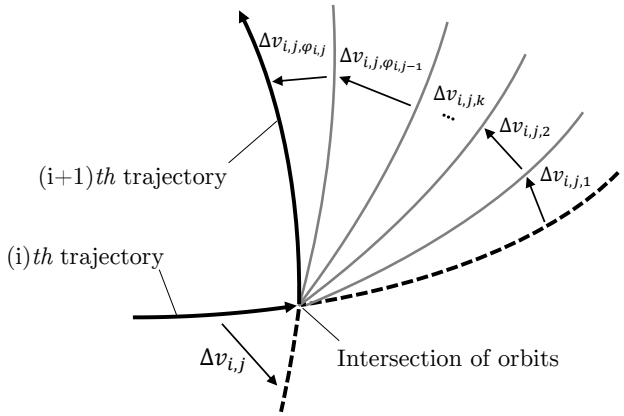

Fig. 2.4: Impulse division at intersection of orbits.

where $\Delta v_{i,j,k}$ is the kth minor impulse in the jth intersection of orbits in the ith stage. The current division keeps the impulse direction fixed as $\Delta \mathbf{v}_{i,i,k}/|\Delta \mathbf{v}_{i,i,k}| = \Delta \mathbf{v}_{i,j}/|\Delta \mathbf{v}_{i,j}|$ and it satisfies $\Delta v_{i,j,k} = \Delta v_{i,j}/\varphi_{i,j}$ if the desired impulses are equally divided. Calculating the orbital elements based on the newly obtained velocity vectors and rearranging them back to the discretized Lambert problem in the previous section will end in the solutions satisfying the given impulse limit. The solution obtained here will be used as a candidate solution to be utilized by the developed EA in the following section. Considering the fact that the EA starts with a random distribution of individuals within the known boundary limits, to minimize the overall transfer time, one can take advantage of univariate Gaussian distribution and sample individuals near the fuel-optimal region for seeding the algorithm instead of uniform random distribution.

2.4 Hybrid Self-Adaptive Evolutionary Algorithm

The algorithm for solving the multi-impulse discrete Lambert approach is described in this section. Particularly, a hybrid self-adaptive algorithm that combines an enhanced evolutionary algorithm with the feasible solution is presented. During the development of the EA, the features of the space rendezvous mission are utilized to make the approach robust to any changes in the space mission. In this section, three components are presented briefly, describing the main enhancements that have been taken into account. The aim of this section is to present a layout of the connection between the algorithm parameters and the elements of space rendezvous, which are utilized to make the algorithm self-adaptive. Details regarding each modification are omitted and the reader is urged to refer to the references provided.

2.4.1 Hybridization of Algorithms

According to the presented approach, the search space of the discretized Lambert problem is a continuous domain, possibly with a variable number of local optima depending on the type of mission. Since the goal is to reach the global optimal solution considering three objectives in general, there is a high possibility for the solution to get trapped in the local optima regions. In order to take advantage of the swarm intelligence and the shape of the landscape of the problem and also compensate the weaknesses of stochastic and gradient-based methods, hybridization of methods from two different types of algorithms has been taken into account. In recent years, such hybrid evolutionary algorithms have been well developed in different spacecraft trajectory design and optimization problems [155].

The core of the optimization algorithm in this approach is based on an Improved Particle Swarm Optimization (IPSO), hybridized with an NLP solver. IPSO performs searching via a swarm of particles that updates from iteration to iteration considering some enhancements. To seek the optimal solution, each particle moves in the direction to its previously best (*pbest*) position and the global best (*gbest*) position in the swarm [16]. One has

$$pbest(i,j) = \underset{k=1,\ldots,j}{\arg\min}[J(x_i(k))], \quad i \in \{1,2,...,N_p\} \tag{2.9}$$

$$gbest(j) = \underset{\substack{i=1,\ldots,N_p \\ k=1,\ldots,j}}{\arg\min}\,[J(x_i(k))] \tag{2.10}$$

where i here denotes the particle index, N_p the total number of particles, and j the current iteration number. The velocity $\hat{v}$ and position $\hat{p}$ of particles are updated by the following equations:

$$\hat{v}_i(j+1) = w_i(j)v_i(j) + c_1\delta_1(pbest(i,j) - x_i(j)) + c_2\delta_2(gbest(j) - x_i(j)) \tag{2.11}$$

$$x_i(j+1) = x_i(j) + \hat{v}_i(j+1) \tag{2.12}$$

where $\hat{v}$ denotes the velocity, $w_i(j)$ is the inertia weight used to balance the global exploration and local exploitation, δ_1 and δ_2 are uniformly distributed random numbers within range $[0,1]$, and c_1 and c_2 are personal and global learning coefficients.

The inertia weight $w_i(j)$ is to bring about a balance between the exploration and exploitation characteristics of the process. A large inertia weight facilitates a global search while a small inertia weight facilitates a local search. By changing the inertia weight dynamically, the search capability is dynamically adjusted. In this algorithm, $w_i(j)$ is defined as [156].

$$w_i(j+1) = \begin{cases} min\Big(1, w_i(j) + (1-w_0) \times \big(\exp\big(\frac{(xi(j+1) - pbest(i,j))^2}{-2\sigma^2}\big) + \rho\big)\Big) & \text{if } \delta_i(j) > 0 \text{ and } \delta_i(j+1) > 0 \\ max\Big(0.1, w_i(j) - w_0 \times \big(1 - \exp\big(\frac{(xi(j+1) - pbest(i,j))^2}{-2\sigma^2}\big) - \rho\big)\Big) & \text{else if } \delta_i(j) < 0 \text{ and } \delta_i(j+1) < 0 \\ w_i(j) & \text{otherwise} \end{cases} \tag{2.13}$$

where w_0 is the initial inertia weight which is considered equal for all particles in all dimensions, in the $(j+1)th$ iteration. The Gaussian kernel width (σ) is adjusted in a way that covers the maximum movement of the particles. ρ is a small positive number used to ensure a proper increase or decrease of the inertia weight. δ_i is the feedback parameter defined as:

$$\delta_i(j+1) = \begin{cases} 1 & \text{if } J(x_i(j)) < pbest(i, j-1) \\ -1 & \text{else} \end{cases} \tag{2.14}$$

According to this equation, the last two steps during the course of the run are analyzed to be sure of making decisions about the value of the inertia weight. When a particle succeeds in some sequential steps, it will have more tendency to memorize its direction. Probably because it will have more success in this direction. When a particle does not succeed in some sequential steps, it has less tendency to memorize its previous direction. Probably because it will have no more success in this direction.

In every generation within the process of optimization, the position of the best particles $x_i(j)$ is improved with an NLP, leading to fast convergence. LBFGS [157], an approximation to BFGS, which requires a lot less memory is used as an efficient NLP for this matter. The position of the ith particle is improved in the jth iteration as:

$$x_i(j+1) = x_i(j) - \alpha_j H_j \nabla J(x_i(j)) \tag{2.15}$$

where α_j is the step length and H_j is updated at every iteration by means of the formula

$$H_{j+1} = V_j^T H_j V_j + \rho_j s_j s_j^T \tag{2.16}$$

where

$$\rho_j = \frac{1}{y_j^T s_j} \tag{2.17}$$

$$V_j = I - \rho_j y_j s_j^T \tag{2.18}$$

and

$$s_j = x_i(j+1) - x_i(j) \tag{2.19}$$

$$y_j = \nabla f_{j+1} - \nabla f_j \tag{2.20}$$

Since the inverse Hessian approximation H_j will generally be dense, the cost of storing and manipulating it is prohibitive when the number of variables

is large due to high number of impulses in the rendezvous mission. To circumvent this problem, a modified version of H_j is utilized implicitly, by storing a certain number (λ) of the vector pairs (s_j, y_j) used in the formulas Eq. 2.16 to Eq. 2.20. The product $H_j \nabla J(x_i(j))$ can be obtained by performing a sequence of inner products and vector summations involving $\nabla J(x_i(j))$ and the pairs (s_j, y_j). After the new iterate is computed, the oldest vector pair in the set of pairs (s_j, y_j) is replaced by the new pair obtained from the current step in Eq. 2.19 and Eq. 2.20. In this way, the set of vector pairs includes curvature information from the most recent iterations. Practical experience has shown that modest values of λ often produce satisfactory results. The algorithm also benefits from other modifications as well, such as mirror effect and velocity clamping. The reader may refer to [158, 159, 160] for the details.

2.4.2 Generating Near-optimal Transfers

The proposed analytical seeding is utilized for generating the populations for the developed EA. In order not to lose the diversity of the populations, uniform random distribution of individuals is also utilized alongside the Gaussian random distribution to reach the optimal transfer. The new population x_n is then generated as a vector of mixed individuals. Assuming the generation of n individuals (x_n), the algorithm generates $\epsilon \times n$ individuals based on the uniform distribution (x_u) and produces $(1-\epsilon) \times n$ individuals based on Gaussian distribution (x_g) near the region in the search space which has the same minimized-fuel as in the feasible solution as:

$$x_g \sim \mathcal{N}(x^*, \sigma^2(p_0, p_f, N)) \tag{2.21}$$

where x^* is the feasible solution with the minimized fuel the same as the solution with N_0 stages, which satisfies the impulse limit and σ^2 is the selective variances. One example of such a distribution is illustrated in Fig. 2.5.

Fig. 2.5 shows the intermediate orbits in a four-impulse rendezvous problem, separated in the two distribution types mentioned. The variance of the Gaussian distribution σ^2 is a vector as $\sigma^2 = [\sigma_a^2; \sigma_e^2; \sigma_i^2; \sigma_\Omega^2; \sigma_\omega^2; \sigma_\theta^2; \sigma_t^2]$, representing different variances for each type of variables in the optimization. Regarding the proposed approach, the input vector x contains the orbital elements of the intermediate transfer trajectories along with the Lambert problem variables in each stage. Since the scale of variables in the input vector is different, the variances should be selected properly as constant values, or as the functions of some characteristics from initial orbit p_0 and final orbit p_f in the space rendezvous mission.

2.4.3 Automatic Parameter Tuning

Since the objectives J_f, J_v and J_t have different types and scales, ζ and ξ should be carefully tuned for each rendezvous problem. Also, the variance vector of variables σ^2 used in Eq. 2.21 needs to be tuned according to a feedback

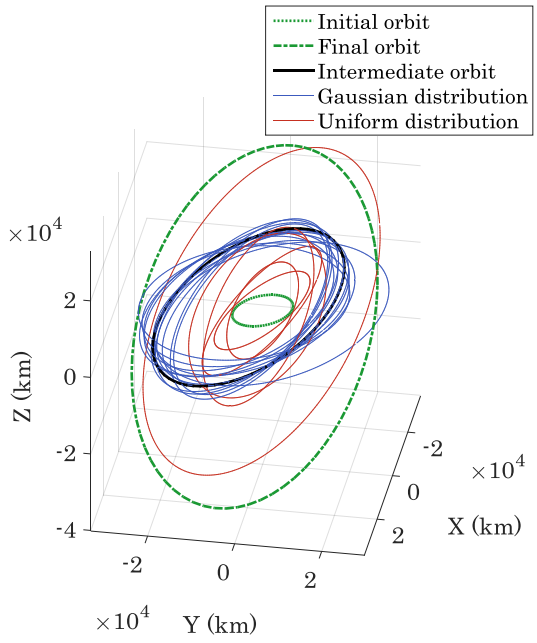

Fig. 2.5: Distribution orbits in four-impulse rendezvous ($\epsilon = 0.4$).

variable from the rendezvous problem. Rather than the seeding technique and the optimization algorithm, the parameters ζ and ξ are also tuned as functions of the rendezvous problem itself. Obviously, one selection of these parameters for a specific rendezvous mission does not necessarily result in the optimal solution in another mission. The issue is that depending on the shape and orientation of the initial and final orbits or in general the amount of difference between the orbital elements (p_0 and p_f), the sensitivity of these objectives varies. For instance, these parameters should be somehow auto-tuned to prevent the algorithm from sacrificing the impulse violation in favor of time. If this happens, the impulses achieved are not feasible, regardless of the total transfer time. Regarding the impulse violation coefficient ζ, an arbitrary parameter Γ, representing the difference of initial and final orbits as $\Gamma(p_0, p_f)$ is defined by the following formula:

$$\Gamma = \frac{|a_0 - a_f|}{3Re} + \frac{|e_0 - e_f|}{0.5} + \frac{|i_0 - i_f|}{\pi} + \frac{|\Omega_0 - \Omega_f|}{\pi} + \frac{|\omega_0 - \omega_f|}{\pi} \tag{2.22}$$

where Re is the Earth radius. This formula has five terms, corresponding to each orbital element, divided by some scaling factors. These scaling factors are selected in order to have the same impact on the overall value of Γ. The proof of this claim is illustrated in Fig. 2.6 and Fig. 2.7.

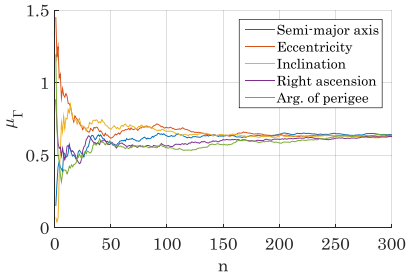

Fig. 2.6: Cumulative mean value of Γ terms.

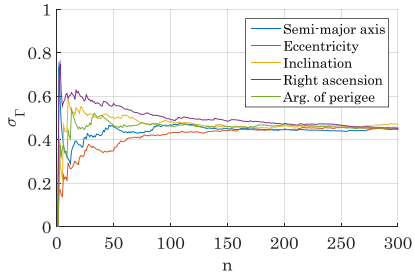

Fig. 2.7: Cumulative standard deviation of Γ terms.

Fig. 2.6 and Fig. 2.7 show the cumulative mean value μ_Γ and the standard deviation σ_Γ of the related term in Γ for n instances of different pairs of space orbits as the rendezvous missions. The plots show that the scaling factors are adjusted fairly for uniform changes of every Γ terms regarding each orbital element as they converge almost to a same value. Besides the difference of orbital elements, the minimum number of impulses required for the transfer according to the impulse limit is considered as a tuning feedback. This parameter, denoted by ϕ, can be calculated after generating the feasible solution described in Section 2.3. Having the required impulses at orbit intersections $\varphi_{i,j}$, the total number of impulses is as:

$$\phi = \sum_{i=1}^{N} \sum_{j=1}^{2} \varphi_{i,j} \tag{2.23}$$

where N is the total number of stages. The reason for considering this variable as feedback for tuning the algorithm parameter is that when the number of impulses increases, the likelihood of returning infeasible solutions by the algorithm will be higher. Utilizing this parameter, the variance vector of decision variables used in Eq. 2.21 is tuned as:

$$\sigma^2 = \begin{bmatrix} \sigma_a^2 & \sigma_e^2 & \sigma_i^2 & \sigma_\Omega^2 & \sigma_\omega^2 & \sigma_\theta^2 & \sigma_t^2 \end{bmatrix}' = \chi(1 - e^{-\beta\phi}) \tag{2.24}$$

where vector χ includes coefficients for each type of optimization variables. One selective value for χ is 0.1 for eccentricity, 180 for true anomalies, 5000 for time, and 100 for the rest of the optimization variables. Also, the value of 0.1 for β adjusts the variances into a more comprehensive amount. Such a selection scales the variances to have the optimum distribution of the generated near optimal solutions. Also, regarding the obtained variables, the parameter ζ is defined as:

$$\zeta = \Gamma + \ln(\phi) \tag{2.25}$$

This tuning method adjusts the impulse violation according to the complexity of the space mission and the number of stages which itself varies according to

the given impulse limit. The coverage of this modeling is illustrated in Fig. 2.8.

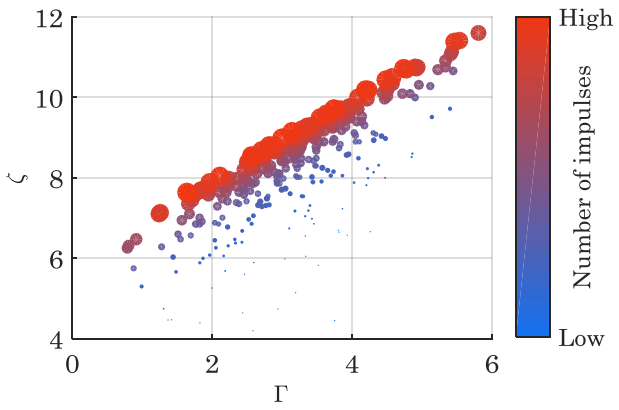

Fig. 2.8: Impulse violation weighting coefficient coverage.

Fig. 2.8 shows the variation of ζ corresponding to the cost of impulse violation for the instances mentioned. A random impulse limit is considered for each space rendezvous mission, leading it to produce a different number of impulses for every long-range rendezvous. The points referring to high number of impulses are shown as big markers while smaller markers refer to the space missions with a low number of impulses. The proposed method makes the cost of impulse violation comparable to the fuel cost with respect to the complexity of the rendezvous mission and the impulse limit. Similarly, the weighting coefficient for the transfer time is defined as:

$$\xi = (J_t^*)^{-1} \tag{2.26}$$

where J_t^* is the overall maneuver time of the multi-impulse transfer obtained within the analytical seeding phase of the algorithm. Since the algorithm attempts to find time-optimal solutions from the fuel-optimal region of the solution domain within the process of optimization, it is cruicial to have a dynamic coefficient for this cost with respect to the maximum transfer time between the initial individuals. Considering this value as the weighting coefficient for the time scale, the overall time will be scaled to one with respect to the initial seeds, making the value of J_t comparable to the rest of the costs.

Once all of the parameters have been tuned, the proposed EA can be utilized to search for the best solution. The pseudocode for this strategy is presented in Algorithm 1.

As shown, the proposed strategy in dealing with the multi-objective space rendezvous ends up facing an optimization problem while having candidate

Algorithm 1: Long-range rendezvous optimization algorithm

Input: p_0, p_f, η
Data: $SeedingMethod, DiversityCheck, NLPMethod$
$N_0 \leftarrow 1$
$\zeta_0 \leftarrow 0$
$\xi_0 \leftarrow 0$
Construct problem **From** $[p_0, p_f, N_0, \zeta_0, \xi_0]$
$solution \leftarrow Solve(problem)$
Extract $\Delta v_{i,j}$ **From** $solution$
if $\Delta v_{i,j} \leq \eta$ **then**
  Extract X **From** $solution$
else
  Calculate $\varphi_{i,j}(\Delta v_{i,j}, \eta)$
  Calculate $\phi(\varphi_{i,j})$
  Calculate $\Gamma(p_0, p_f)$
  Calculate $\zeta(\Gamma, \phi)$
  Extract J_t^* **From** $solution$
  Calculate $\xi(J_t^*)$
  Calculate $\sigma^2(\chi, \beta, \phi)$
  if *SeedMethod = "Gaussian"* **then**
    Extract X **From** $solution$
    Construct X_0 **From** $\mathcal{N}(X, \sigma^2)$
  else if *SeedMethod = "Uniform"* **then**
    Construct X_0 **From** $\mathcal{U}$
  end if
  Calculate $N(\phi)$
  $nPop \leftarrow 10N$
  $nGen \leftarrow 20N$
  Construct problem **From** $[p_0, p_f, N, \zeta, \xi]$
  for $i \leftarrow 1$ **to** $nGen$ **do**
    Update w_i
    $X_i \leftarrow PSOiter(problem, X_{i-1})$
    if *NLPMethod* **then**
      $X_i \leftarrow SolveNLP(problem, X_i, NLPMethod)$
    end if
    if *DiversityCheck* **then**
      $X_i \leftarrow RefinePop(X_i, \sigma^2)$
    end if
  end for
  $X \leftarrow X_i$
end if
Result: X

solutions near the fuel-optimal region that satisfies the impulse limit. If the diversity of the populations is not satisfactory, such solutions are regenerated during the optimization process. Also, the parameters involved in the process are automatically tuned based on the orbital elements of the orbits and the required impulse limit.

2.5 Numerical Simulations

The proposed approach is investigated in several aspects in this section. First, a sample space rendezvous is solved considering two various impulse limits and the obtained orbital maneuvers are analyzed. The performance of the approach in finding the best minimum transfer time while having the near optimal fuel consumption without violating the impulse limit is studied when different impulse limits are considered for the same space rendezvous. Next, experiments are performed in which the approach is utilized in many different space rendezvous missions. Following the experiments, the performance of the proposed algorithm is compared with other standard EAs, indicating the superiority of the proposed algorithm due to the enhancements.

2.5.1 Long-range Rendezvous

A space rendezvous mission with two different impulse limits is evaluated. Consider an orbit to orbit rendezvous as in Table 2.1.

Table 2.1: Orbital elements of the orbit to orbit rendezvous

Orbital elements	Initial	Final
a (km)	11300	32600
e	0.2	0.5
i (deg)	40	50
Ω (deg)	275	270
ω (deg)	280	265

This space rendezvous is a non-coplanar transfer problem and two scenarios are considered with impulse limits of $\eta = 200m/s$ and $\eta = 50m/s$. As for the first step, disregarding the impulse limit and the transfer time, the best solution found for the optimal two-impulse transfer with minimum Δv is the one that starts at $\mathbf{r}_i = [-3887; -7694.2; -3811.9]$ km on the initial orbit and finishes at $\mathbf{r}_f = [4454.4; 25862; 5308.5]$ km on the final orbit. This solution is obtained with the NLP method described in the previous section with few iterations. This transfer takes 24157 seconds with the total Δv of $2.1353km/s$. Considering this solution as the fuel-optimal transfer, the analytical multi-impulse transfer is extracted and seeded to the developed hybrid

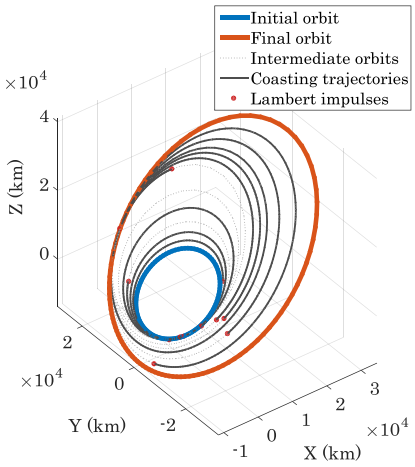

Fig. 2.9: Optimized multi-impulse orbit rendezvous ($\eta = 200m/s$).

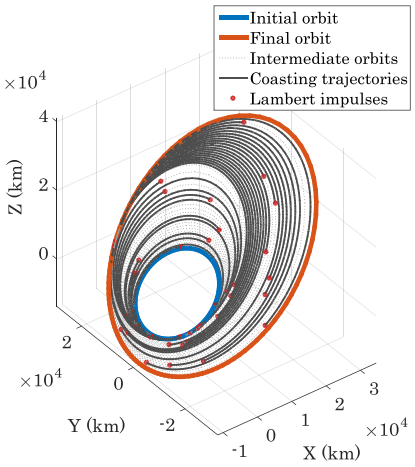

Fig. 2.10: Optimized multi-impulse orbit rendezvous ($\eta = 50m/s$).

self-adaptive algorithm. The algorithm parameters are tuned automatically based on the multi-impulse solution and the orbital parameters of the initial and final orbits as tabulated in Table 2.2.

Table 2.2: Auto-tuned parameters of the self-adaptive algorithm

	ϕ	ζ	ξ	σ_a^2	σ_e^2	σ_i^2	σ_Ω^2	σ_ω^2	σ_θ^2	σ_t^2
$\eta = 0.20$	12	4.36	2.40×10^{-6}	77.79	0.0419	3.88	1.94	5.82	125.79	3494
$\eta = 0.05$	44	5.66	7.27×10^{-7}	109.95	0.0592	5.48	2.74	8.23	177.79	4938

The impulse discretization is performed at the intersections. Having the impulses at each intersection, the required number of divisions (φ) are calculated using Eq. 2.7. It gives 6 divisions at each intersection for the first case and 23 and 21 divisions for the second case based on the desired impulse limit. Based on the obtained number of divisions in each intersections, the total number of impulses ϕ are achieved for each case using Eq. 2.23. The parameter Γ representing the difference of orbits is 1.8799 for this transfer based on Eq. 2.22. Then, the scalarization coefficient ζ is obtained from Eq. 2.25 for both cases. Besides, the variances of optimization variables in σ^2 vector can be obtained from Eq. 2.24 Also, division of the impulses yields the multi-impulse transfers with the overall mission duration of 4.8214 days and 15.905 days for $\eta = 200m/s$ and $\eta = 50m/s$ respectively. By having these transfer times from the initial solution, the scalarization coefficient ξ can be calculated from Eq. 2.26. Having all of the tuned parameters, the main problem can be tackled by the presented approach. The best time-optimal solutions are obtained with

respect to the tuned parameters utilizing the developed self-adaptive algorithms. 3D visualization of the obtained transfers are depicted in Fig. 2.9 and Fig. 2.10.

In these maneuvers, the spacecraft travels between the stages which are optimized to have minimum overall transfer time, resulting in different Lambert problems in each revolution. The sequence of intermediate orbits along with the solution of the Lambert problem in each stage is optimized in order to minimize the overall transfer time. As shown in the figures, a lower impulse limit generally yields longer maneuver with more revolutions and impulses. The variations of impulses for both maneuvers are depicted in Fig. 2.11.

As shown, the impulse limits are satisfied for both maneuvers, leading to the conclusion that the weighting coefficients are almost properly tuned using the proposed method. Usually the untuned parameters lead the algorithm to sacrifice the impulse limit in the favor of transfer time which is not desired in the current concept. The optimized variables, such as solutions to the Lambert problems, can be plotted for either case. As previously mentioned, the problem input vector includes the true anomalies of the points where the spacecraft travels between two sequential stages. Their optimized values for two cases are illustrated in Fig. 2.12 and Fig. 2.13.

In these figures, the optimized anomalies are plotted during the overall orbital maneuver. Small points refer to the jumps at the beginning of the maneuver while bigger points are related to the impulses near the end of the mission. Comparing the location of the optimized true anomalies in the plots for two cases indicates that two potential regions can be identified as the near-optimal region for true anomalies between Lambert problems.

Regarding the obtained solution, the time histories of orbital elements can be simulated for the spacecraft as it travels from the initial orbit to the final orbit. The variation of orbital elements are shown in Fig. 2.14 to Fig. 2.17. Several observations can be inferred from the time histories of orbital

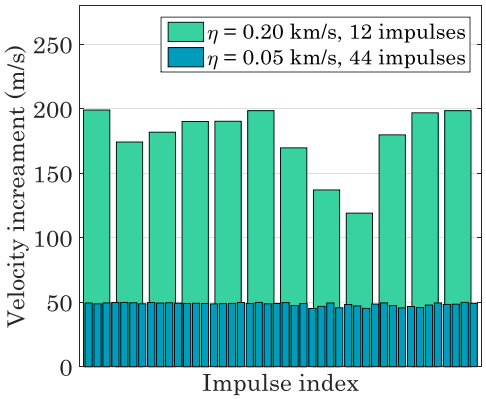

Fig. 2.11: Sequence of impulses in multi-impulse long-range rendezvous.

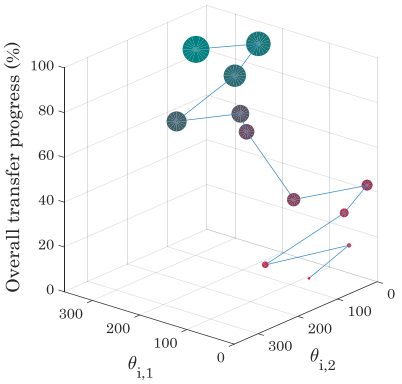

Fig. 2.12: True anomalies of the Lambert problem ($\eta = 200m/s$).

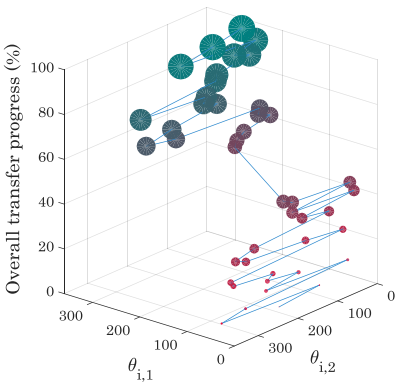

Fig. 2.13: True anomalies of the Lambert problem ($\eta = 50m/s$).

elements. The non-linear variation of the elements shows that considering the linear variation of elements is generally not an optimal choice, which is a confirmation for the practicality of the proposed method as it is actually a shape-based approach. Another observation is the time distance between two sequential impulses in the results. According to the variation of elements, this time distance increases as the spacecraft reaches the final orbit in both cases. This increment is in agreement with the variation of $a(t)$. Since the semi-major axis of the initial orbit is less than the final orbits, the time of one revolution will increase as the spacecraft travels between the intermediate orbit. This is due to the fact that the orbital period depends only on the semi-major axis in each stage.

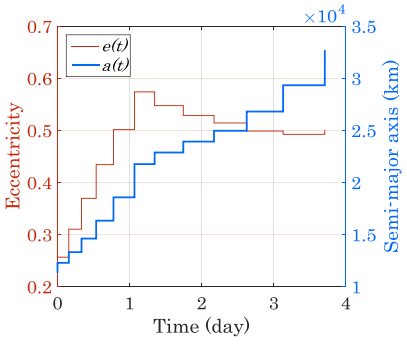

Fig. 2.14: Time histories of semi-major axis and Eccentricity ($\eta = 200m/s$).

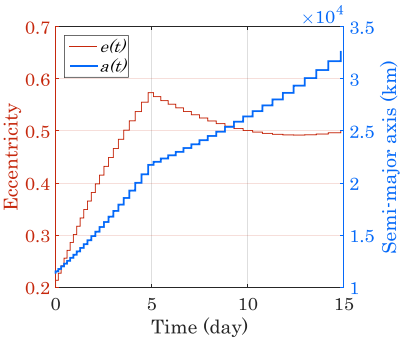

Fig. 2.15: Time histories of semi-major axis and Eccentricity ($\eta = 50m/s$).

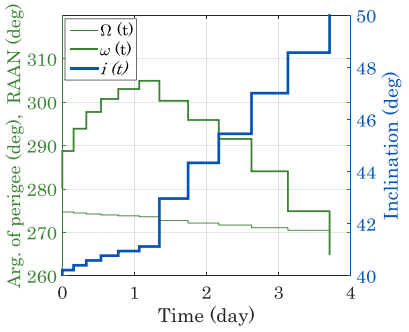

Fig. 2.16: Time histories of inclination, Arg. of Perigee and RAAN ($\eta = 200m/s$).

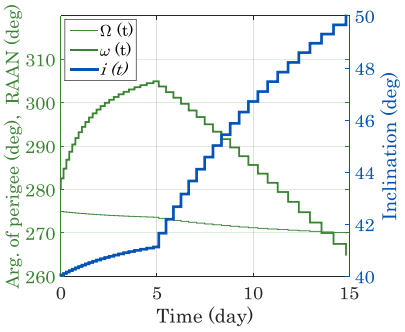

Fig. 2.17: Time histories of inclination, Arg. of Perigee and RAAN ($\eta = 50m/s$).

The solutions can be also analyzed when various impulse limits are considered for a space rendezvous. As the impulse limit decreases, the tendency of the problem shifts to complicated solution domains, which are more difficult to optimize. Table 2.3 shows the characteristics of the previously defined space rendezvous problem considering various impulse limits.

In this Table, the entire process of tuning the parameters, generating near-optimal feasible solutions, and solving the problem is performed for each impulse limit. First, the proposed simple feasible solution is obtained initially for each impulse limit, and the transfer times that were obtained before using the main algorithm are presented in the table. Then, the main algorithm is utilized to find the best solution based on the initial feasible solution. The obtained solutions are presented in the table to make a comparison regarding the difference between the transfer times. According to the results from the optimizations, the absolute improvement of the objective related to transfer time varies from case to case. Fig. 2.18 illustrates a representation of the best solution found so far regarding each impulse limit.

Table 2.3: Characteristics of space rendezvous problems.

Impulse limit (m/s)	300	250	200	150	100	75	50	25
Optimization variables	59	75	91	115	179	235	347	691
Initial transfer time (day)	4.088	4.536	4.821	6.899	10.304	13.123	15.906	36.922
Best transfer time (day)	2.301	2.992	3.710	4.673	7.466	9.854	14.828	29.701

In this representation, the best 1000 solutions are saved during every optimization and are plotted altogether to give an insight to the near-optimal region. For each unique impulse limit, the best solution is considered as the target and the relative distance of the rest of the solutions, denoted by d_i, is

computed as:

$$d_i = \frac{x_i - x^*}{\max\limits_{i \in \{1,\dots,n\}} [x_i - x^*]} \tag{2.27}$$

where n is the number of solutions, i is the index of each solution, x_i is the input vector of each solution, and x^* is the best solution so far. Having, d_i vector for each solution, the euclidean distance for that solution, denoted by D_i, can be computed as:

$$D_i = \sqrt{\sum_{j=1}^{k} d_i(j)^2} \tag{2.28}$$

where k here is the size of the input vector which varies from case to case in the current analysis. Regarding the distances, the two local optima regions can be identified, which is very near the global best solution, leading to conclusion that the optimization process successfully identified the solution within the actual near-optimal region. As the impulse limit decreases, the absolute value of the overall transfer time increases. However, it does not have a linear behavior with respect to the given limits. The distribution of points also indicates that the gradient of the solution domain within the near-optimal region is higher in the low impulse limit in comparison to the high impulse limit, confirming that the problem is more difficult to deal with in the first case. The absolute amount of solution improvement within the selected solutions is different regarding the results. In order to have an insight about the relative percentage of the improvement, the scaled values of the objectives convey more comprehensible data regarding the percentage of the improvement. One representation of the relative improvement of the selected solutions is shown in Fig. 2.19.

Here, τ is the relative objective improvement calculated by:

$$\tau = \frac{J^* - J}{J^*} \tag{2.29}$$

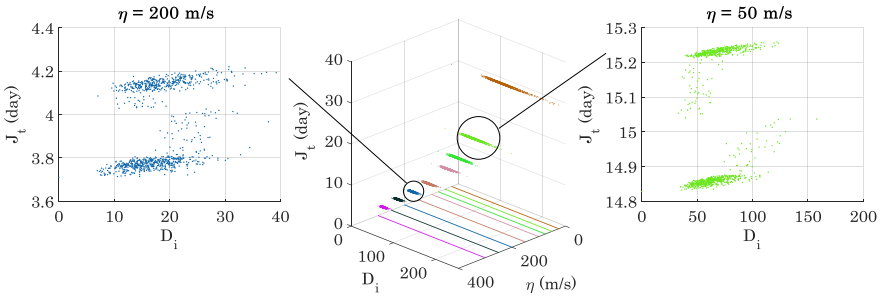

Fig. 2.18: Euclidean distances of the best solutions in minimum-time minimum-fuel rendezvous.

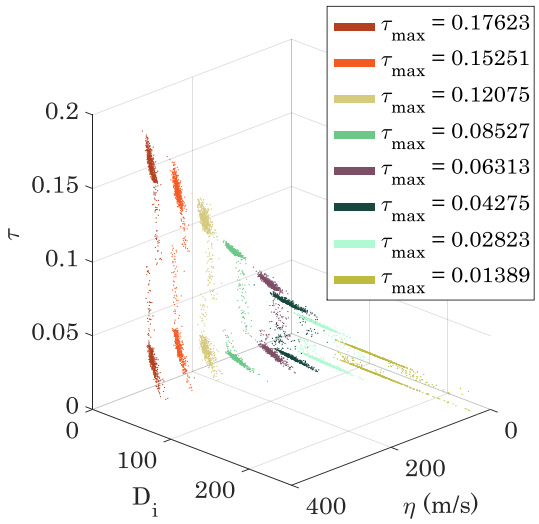

Fig. 2.19: Relative objective improvement for various impulse limits.

where J_t^* is the total transfer time obtained by the simple feasible solution. Comparing the τ values indicates that, although the absolute amount of the improved objective in impulsive rendezvous with very low impulse limit is high, the relative improvement is lower than those with higher impulse limits. This is the observation which confirms that the complexity of the optimization problem significantly increases and less improvement is gained by the algorithm as more impulses are taken into account for the space rendezvous.

As stated, the algorithm utilizes the analytical solution in two ways. First is the initial seeding in which the initial population is generated based on near-optimal region according to the obtained feasible solution. The second one is a trigger during the optimization process which eliminates similar individuals and regenerates them based on a method the same as the initial seeding. The initial seeding of the algorithm and regeneration of individuals during the optimization process is handled by ϵ as described in this approach. This parameter specifies the balance between the Gaussian distribution and the uniform distribution when generating new individuals. The quality of the final solution achieved by the algorithm depends on the choice of ϵ. Comparing the best solutions achieved with different values of ϵ considering various impulse limits is a matter of interest as it shows the rather optimal value of these parameters in different rendezvous problems. An illustration of such a comparison is shown in Fig. 2.20.

In this analysis, the problem is solved for each impulse limit considering different values for ϵ. In each case, the optimization is carried out 10 times and all the objective improvements (τ_{max}) are stored. Then, the average of improvement is calculated as:

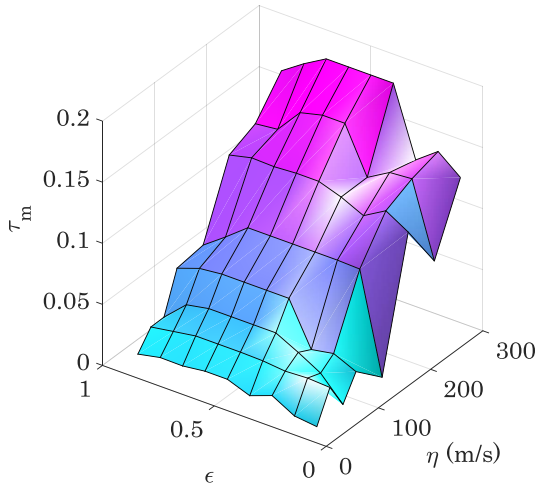

Fig. 2.20: Average performance of the algorithm in 20 runs.

$$\tau_m = \frac{1}{n_r}\sum_{i=1}^{n_r}\tau_{max}(i) \tag{2.30}$$

where n_r is the total number of optimization runs for each case, $\tau_{max}(i)$ is the improvement in the ith run. The surface illustrating the average performance of the algorithm shows that typically a value of 0.4 to 0.6 results in almost the maximum performance for all of the cases. It can also be concluded that in the rendezvous missions with higher impulse limits, the performance is affected more by the ϵ in comparison to missions with low impulse limits.

2.5.2 Empirical Experiments

When comparing one metaheuristic to another in a spacecraft trajectory optimization problem, it is crucial to perform benchmark tests using a suite of standard problems [2]. In order to make a practical comparison between algorithms, it is important to consider several factors. These factors include testing a large suite of instances of the rendezvous problems, using the same initial guess, and comparing them based on the same convergence criteria. To achieve this end, 100 instances of orbit to orbit rendezvous are tested by the proposed approach and some standard evolutionary algorithm. The orbital elements of the initial and final orbits are plotted in Fig. 2.21.

According to Fig. 2.21, the orbital elements are uniformly distributed as $6600km < a_i, a_f < 50000km$, $0 < e_i, e_f < 0.8$, $0^\circ < i_i, i_f < 90^\circ$, $0^\circ < \Omega_i, \Omega_f < 360^\circ$ and $0^\circ < \omega_i, \omega_f < 360^\circ$. The elements are considered to satisfy the conditions as $-5000km < \Delta a < 5000km$, $-0.4 < \Delta e < 0.4$, $-30^\circ < \Delta i < 30^\circ$, $-30^\circ < \Delta\Omega < 30^\circ$, $-30^\circ < \Delta\omega < 30^\circ$. A random impulse limit is

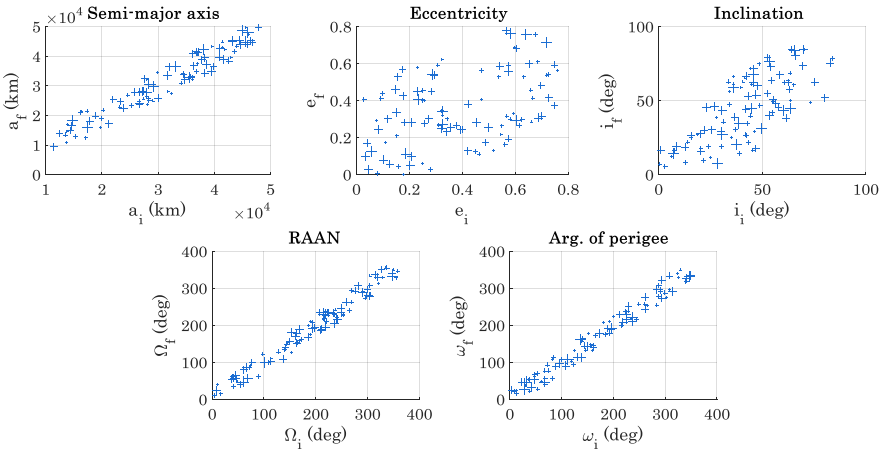

Fig. 2.21: Distribution of orbital elements for empirical experiments.

dedicated to each space mission within the range of $50m/s < \eta < 500m/s$, resulting in problems with different number of variables and complexities. In Fig. 2.21, larger impulse limits are plotted with big markers while smaller ones represent space rendezvous missions with lower impulse limits.

The algorithm presented has many features for robustness and self adaptiveness including hybridization with NLP, adaptive weighting coefficient, regeneration of individuals based on near fuel-optimal region, dynamic damping ratio, and analytical seeding based on discretization of orbit intersections. Therefore, numerous versions of this approach can be implemented by enabling or disabling any of these features. They will have a different effect on the obtained solution. However, due to brevity, only three types of the approach are implemented and compared with other EAs. The first one is the Hybrid Self-Adaptive Evolutionary Algorithm (HSAEA), which is the full developed algorithm with all of the improvements. The second is the Self-Adaptive Evolutionary Algorithm (SAEA-I), which is the same as HSAEA but no hybridization with NLP is applied during the optimization process. The third one is another reduced version of the Self-Adaptive Evolutionary Algorithm (SAEA-II), which is the same as SAEA-I, but no regeneration of individuals for diversity correction is used during the optimization process. On the other hand, all these three versions benefit from the tuning of the parameters based on the characteristics of the rendezvous problem and dynamic damping ratio. Standard Particle Swarm Optimization (PSO) and Genetic Algorithm (GA) are also taken into account for comparison.

Since each space rendezvous mission with a distinct impulse limit generates a different optimization problem, the problems will have different complexities and a different number of optimization variables due to the various necessary numbers of impulses. Therefore, the common parameters of the al-

gorithms have been chosen in such a way that a fair comparison between the performance of each algorithm is achieved. For each problem, the number of populations and the number of generations for all algorithms are set as $10n$ and $20n$ respectively, where n is the number of optimization variables for that space rendezvous mission. Regarding the three versions of the proposed approach, HSAEA, SAEA-I and SAEA-II, all algorithm parameters are tuned automatically for each problem according to the presented process in Section 2.4. For PSO, the values of 1.8 and 2.0 are chosen for personal and global learning coefficients respectively. Also, the inertial weight is set to 1.0 with the damping ratio of 0.95 per generation. For GA, crossover percentage and crossover range factor are chosen as 0.6 and 0.3 respectively. Also, the mutation percentage and the mutation range are selected as 0.4 and 0.2 respectively. The reason for choosing these parameters for PSO and GA is that they are statistically shown to have the best performance for these algorithms in most of the space rendezvous missions with the current setup. Since the proposed algorithms benefits from the automatic tuning of parameters, the best settings of the two selected EAs are chosen for performance comparison. Although tuning the algorithm parameters is another optimization problem itself, the effort in this research is to use the best performance of PSO and GA for comparison as these values outperform other combinations in most instances of space rendezvous missions.

For this analysis, HIPATIA cluster setup of BCAM is used with 18 nodes including 624 cores (Processor Intel(R) Xeon(R) Gold 6140 CPU @ 2.30GHz) and 4352 GB RAM. Having 100 space rendezvous problems, each problem is solved with the 5 aforementioned algorithms (HSAEA, SAEA-I, SAEA-II, PSO, GA) and each algorithm is run 10 times. Therefore, a total number of 5000 jobs are submitted to the cluster to run the optimizations in parallel. The solutions obtained are saved for each algorithm and the best solution between all of the runs is assumed to be the global best solution for each rendezvous mission. Then, for each problem, the relative score of the all solutions obtained is calculated as:

$$\gamma_{i,j} = \frac{J_{i,j} - J_{best}}{J^* - J_{best}} \tag{2.31}$$

where $J_{i,j}$ is the objective function obtained for the ith run of the jth algorithm ($0 < i \leq N_r$, $0 < j \leq N_a$), considering 10 runs ($N_r = 10$) for the 5 aforementioned algorithms ($N_a = 5$). J^* is the objective function obtained by the simple feasible solution, and J_{best} is the best obtained solution between all of the runs, which is assumed to be the global best for the rendezvous mission as:

$$J_{best} = \arg\min[J_{i,j}] \quad (i = 1, ..., N_r, j = 1, ..., N_a) \tag{2.32}$$

The reason of considering such a type of score is that different rendezvous missions with various impulse limits will have different values of objective functions, and therefore it is difficult to make a comparison between different

instances. This definition scales the performance of algorithms in each run into a dimensionless score between 0 and 1, where 0 indicates that the algorithm successfully reached the global solution, while 1 shows that the algorithm did not make any improvement over the simple feasible solution and could not decrease the objective function within the optimization process. Any score value in this range shows how much improvement the algorithm obtained in reaching the global best solution. Having all of the scores for the algorithms, an insight to the overall performances can be achieved. Fig. 2.22 shows the performance comparison between the algorithms.

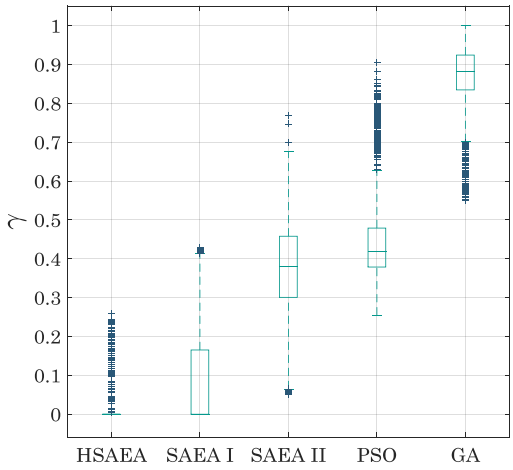

Fig. 2.22: Absolute scores of the algorithms for multi-impulse space rendezvous.

Fig. 2.22 indicates the absolute score of the algorithms applied to different space rendezvous problems. This figure shows the results for 5000 obtained scores (10 runs for each of the 5 algorithms in 100 instances). This score gives an insight into the performance of the algorithms. However, it does not convey the average performance of the algorithms for each instance. The average performance of an algorithm is the mean value of all obtained solutions out of the optimization runs for that algorithm regarding a specific problem. It can be simply calculated as:

$$\gamma'_j = \frac{1}{N_r} \sum_{i=1}^{N_r} \gamma_{i,j} \tag{2.33}$$

where γ'_j is the average performance of the jth algorithm considering all optimization runs. The relative performances of the algorithms based on this score are illustrated in Fig. 2.23.

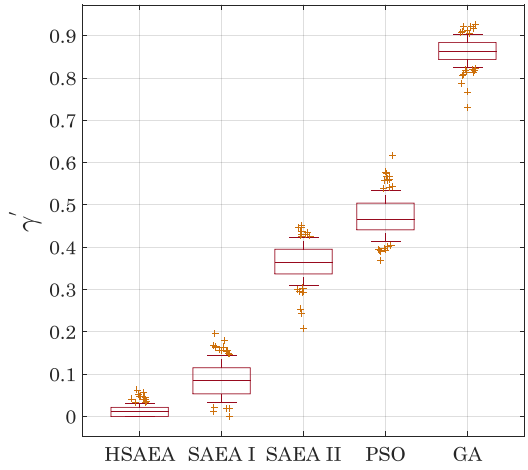

Fig. 2.23: Relative scores of the algorithms for multi-impulse space rendezvous.

Regarding the results, the HSAEA has superior advantage over the rest of the cases, showing the effectiveness of the proposed algorithm when comparing to the standard EAs. Noticing the best scores for each algorithm indicates that only HSAEA and SAEA-I reached the scores of 0 (global best) while SAEA-II, PSO and GA failed to achieve the global best in any of the instances, leading to the conclusion that hybridization of the algorithm with the proposed NLP and regeneration of populations near the fuel-optimal region is necessary to achieve the global best. The comparison between HSAEA and SAEA-I also shows that both algorithms most likely have reached the global optimal solution in the majority of space rendezvous missions and the difference is the probability of reaching the optimal solution. Therefore, it can be implicitly concluded that hybridization with NLP increases the rate of reaching the global optimal solution in different optimization runs, while regeneration of populations near the fuel-optimal region within the optimization process guarantees the convergence to the global optimal point. Comparing the results from SAEA-II and PSO shows that, although the general performance of two algorithms is almost the same, the SAEA-II has a slight advantage in some of the cases over PSO due to dynamic damping ratio. Also, the standard PSO itself outperforms the standard GA in all of the cases, leading to the conclusion that hybridization of GA with the proposed NLP probably does not outperform the presented HSAEA since the proposed HSAEA is generally based on an improved PSO.

While the effectiveness and reliability of the algorithms are compared by showing how close the algorithms get to the best solution, their efficiency can be evaluated by comparing their running time. For the sake of completeness in this research, the speed of the algorithms is measured and compared in

ten selective instances of space rendezvous missions. Results are tabulated in Table 2.4.

Table 2.4: Comparison of dimensionless running time (running time relative to the one associate with GA $[s/s_{(GA)}]$) of the algorithms, omitting the time of cost function evaluations.

Instance	HSAEA	SAEA-I	SAEA-II	PSO
1	7.3198	2.0986	1.1328	1.0898
2	6.2147	2.2233	1.1143	1.0165
3	4.2777	2.0138	1.0457	1.0348
4	5.2043	1.2076	1.0404	1.0274
5	5.3826	1.3224	1.2031	1.0966
6	4.3526	3.1734	1.1279	1.0967
7	6.4117	2.4191	1.0880	1.0501
8	7.1940	2.3722	1.1475	1.0825
9	3.3919	2.5298	1.1231	1.0833
10	4.0869	1.7127	1.1404	1.0682
Average	**5.3836**	**2.1073**	**1.1163**	**1.0646**

In this comparison, the running time of GA is considered as the base score for the speed of other algorithms, since GA has the minimum processing time in this set of algorithms. The time of objective function evaluations are neglected and the calculated running time for each algorithm is divided by the base time of GA for each instance. It has been shown that typical PSO has the least running time as it does not involve any modifications described in this research. SAEA-II, which benefits from automatic parameter tuning and dynamic damping ratio during optimization, requires slightly more time compared to PSO. The comparison of speed between SAEA-I and SAEA-II shows the effect of diversity correction within the optimization. Since regeneration of new individuals is triggered when the diversity of the population is not satisfactory, its occurrence is largely due to the randomness of movement of individuals. Finally, the HSAEA has the slowest process as the NLP employment for improving the quality of the individuals requires more iterations and calculations. It can be observed that the running time is significantly high for HSAEA in comparison to the other algorithms. However, it has the highest possibility to end up finding the best possible solution.

2.6 Conclusions

The long-range space rendezvous problem was addressed using the proposed evolutionary discretized Lambert approach in this chapter. In this research, all objectives, including fuel, time and impulse limit were considered. The

proposed approach attempts to find the global optimal solutions which satisfy the impulse limit while minimizing the fuel and time simultaneously. The fuel-optimal region is identified by means of a discretized approach in which the minimum necessary number of impulses is achieved by dividing the impulses within the intersection of trajectories between two sequential Lambert problems. The obtained solution is utilized to generate near-optimal solutions for seeding the optimization algorithm in order to find the optimal sequence of impulses which results in minimum-time minimum-fuel orbit transfer. The developed self-adaptive algorithm is a hybrid method, combined with auto-tuning techniques and an intelligent individual refinement procedure. The approach is used to solve some multi-impulse rendezvous problems. Results confirm the feasibility of the approach and show that it successfully improves the the optimality of the solution in terms of fuel and time without violating the impulse limit. The percentage of optimality is significant when the approach is used with the developed self-adaptive algorithm. Also, the robustness of the approach is tested by applying the proposed method in different types of rendezvous problems. Comparing the obtained results with the output of standard non-adaptive algorithms indicates the superiority of the developed algorithm. It has been shown that the probability of reaching optimal solutions is significantly higher with the current approach in comparison to other algorithms, since the approach benefits from self-adaptive tuning and hybridization. However, the running time burden is relatively high when the NLP is involved with the optimization process. Increasing the optimality of the solution and the robustness of the technique by means of considering more tuning techniques will be the next step in future research.

3

Short-range Space Rendezvous

3.1 Introduction

Successful short-range rendezvous is a key objective in many space missions as it directly affects various space operations, such as the capturing of a malfunctioned satellite, removing space debris, on-orbit refueling, docking, and repairing spacecraft. In this type of mission, the whole relative spacecraft maneuvering process consists of getting the chaser spacecraft from one orbit to a distance with small relative speed near the target spacecraft. During the past few decades, considerable attention has been paid to spacecraft trajectory optimization of short-range rendezvous problems in the literature [2]. However, due to multiple constraints, it is still challenging to design a robust and efficient algorithm such that the chaser can fly along an optimum feasible path and fulfill different mission requirements.

In short-range space rendezvous, the actual limits and considerations such as thrust limit [161], power limit [162], terminal conditions or other criteria [163] makes spacecraft trajectory optimization a problem with *hard* constraints, with respect to the definition by Malan [164]. In contrast to *soft* constraints, the priority in such short-range space rendezvous missions is to find a feasible solution (reaching the target spacecraft) with the best objective value (minimum fuel), while in a problem with *soft* constraints, objectives and constraints have same priorities.

3.1.1 Short-range Space Rendezvous Trajectory Optimization

Various techniques are employed in different research to satisfy constraints in spacecraft trajectory optimization problems. In some cases, such as multi gravity-assisted interplanetary trajectory optimization [165], simplifying the thrust model from continuous thrust to impulsive, means the problem has only upper and lower bounds for decision variables. However, when it comes to facing short-range rendezvous, more thorough efforts are needed as the problem formulation makes the trajectory optimization challenging in the

sense of finding high quality feasible solutions. For instance, in addition to closed-loop control approaches, such as those based on the Lyapunov function [166], one option is to modify the variables or the formulation to satisfy the constraints, similar to shape-based techniques [167, 168]. These approaches are heavily problem dependent, thus they can not be generalized [169]. Besides these options, another approach is to deal with the constraints within the optimization algorithm [170, 171]. This approach is not a robust method when gradient-based algorithms are employed since the convergence totally depends on the initial guess [172]. Alternatively, Evolutionary Algorithms (EAs) are very popular to deal with constraints in short-range rendezvous. In order to approach constrained continuous optimization problems, EAs utilize a variety of techniques to hold either soft or hard constraints and return the best possible solutions. Coello [173] taxonomized these techniques in different categories: penalty functions, multi-objective approaches, specific operators, repair algorithms and hybrid methods. In addition, the author presented the advantages and disadvantages of each technique.

Regarding the penalty functions, recent advances are mainly on adaptive penalty functions [174] and stochastic ranking [175]. Such an approach can be found in various spacecraft trajectory optimization problems, such as aeroassisted trajectory optimization [176], optimal orbit rising [177] and interplanetary trajectories [178]. These methods are highly competitive over the older methods of penalty functions, such as death penalty functions and static/dynamic penalty functions, which commonly require the derivation of good penalty functions or the penalty factor. However, there are still no guarantees that the methods in [174] and [175] end up in feasible solutions. Regarding the multi-objective approaches, the idea is to consider each constraint as a separate objective function and deal with the problem from a multi-objective perspective [179]. This approach mainly involves Pareto ranking mechanisms, which shows relatively good performance over the other methods. However, in some cases they are more time consuming than penalty functions and may not be able to find solutions with high quality. The approaches based on repairing, and/or hybrid techniques, either suffer from generality, as they are problem-dependent, or they are time consuming. An overview of these developed methods indicates that not all of the techniques are appropriate for dealing with short-range rendezvous problems with constraints, where feasibility is the first priority [173].

3.1.2 Motivation and Contribution

Considering the mentioned facts, developing novel methods for handling constraints, which are independent from the problem and are robust to any types of constraints (equality, inequality, linear, non-linear, etc.) is a need in the literature. In this research, several mechanisms for Estimation of Distribution Algorithms (EDAs) [180] are proposed for handling constraints. EDAs are a class of EAs that work based on the probabilistic models [180]. In an EDA, a

probabilistic model is learned at each iteration and new solutions are sampled from that model. The obtained solutions have similar characteristics as those used for learning the model. One of the characteristics of EDAs is to have an explicit description of the promising solutions in terms of probabilistic models. Following this feature, they have a great potential for enhancement toward further improvements. This characteristic is the main motivation in this research and the effort here is to enhance the mechanisms of EDAs for handling constraints in trajectory optimization of short-range space rendezvous mission. In recent years, there have been some efforts in developing probabilistic models that only generates feasible solutions regarding specific types of constraints [181]. However, up to now, the developments were only applicable in combinatorial optimization [182].

Motivated by the above discussion, in this research, an EDA is proposed specially suited to deal with problems with hard constraints. Particularly, it uses a mixture of probabilistic models and try to better model the feasible region of solutions, and a few mechanisms associated with seeding, learning and mapping methods are proposed, which force the algorithm to generate only feasible solutions, while the optimization process goes on. The proposed seeding mechanism tries to provide initial population of feasible solutions for the algorithm. Within the learning mechanism of the EDA, several operators including feasible conserving clustering and outlier detection are proposed. The first operator initializes the mixture of probabilistic models, centered in feasible region of the solution domain, while the latter improves the convergence of the optimization process. Also, within the sampling stage, various heuristic mapping mechanisms are proposed, which guide the optimization process within the feasible region of the solution domain. The proposed algorithm always returns feasible solutions if the initial feasible population is available and therefore has a great potential to be utilized for spacecraft short-range rendezvous problems.

The EDA with the proposed mechanisms is tested on various benchmarks and space rendezvous problems. First, it is applied on two suites of benchmark problems for constrained continuous optimization and its performance is compared with some state-of-the-art algorithms and constraint handling methods. Conducted experiments confirm the speed, robustness and efficiency of the proposed algorithm in tackling various problems with linear and non-linear constraints. Then, it is tested and analyzed in a short-range space rendezvous mission with different initial conditions considering disturbances in elliptical orbits. Also, the quality of the solutions obtained by the proposed algorithm is compared with another method based on implicit Lyapunov function. Conducted experiments confirm the robustness and efficiency of the proposed algorithm in tackling short-range space rendezvous problems.

The remainder of this chapter is organized as follows. Section 3.2 presents spacecraft dynamics for the short-range rendezvous problem along with mathematical modeling of the control variables. In Section 3.3, an overview of the proposed algorithm is presented, where the mechanisms associated with the

seeding, learning and mapping stages are briefly introduced. The next sections describe each of these mechanisms in detail. Section 3.4 is dedicated to the enhanced seeding mechanism. The learning mechanism based on feasibility conservation is elaborated in Section 3.5. In Section 3.6, the mapping mechanism is explained. The results of the numerical experiments for algorithm verification are provided in Section 3.7 along with comparisons with state-of-the-art algorithms and methods. Afterwards, Section 3.8 is dedicated to the simulation results for validation and verification of the proposed algorithm. Finally, Section 3.9 concludes this chapter.

3.2 Problem Formulation

The mathematical formulation of the motion of the chaser spacecraft relative to the target spacecraft and modeling the input vectors are the primary steps for solving the trajectory optimization problem in short-range space rendezvous. In this research, firstly, relative dynamics with disturbances in elliptical orbits is provided. Then, a direct approach based on discretization and interpolation of the thrust vectors is presented.

3.2.1 Relative spacecraft Dynamics

The dynamics models of the relative motion of the spacecraft are diverse and subject to many considerations and assumptions. A comprehensive survey of models of the dynamics of spacecraft relative motion is available in [183]. In this work, Yamanaka-Ankersen state transition matrix [184] is utilized since it is widely accepted as the state-of-the-art mathematical model for propagation of the relative position and velocity in eccentric orbits, considering disturbances [185, 186]. In this model, it is assumed that the relative distance between the chaser and the target is far smaller than the target orbit radius. Considering $\mathbf{r} = [X, Y, Z]$ and $\mathbf{v} = [\dot{X}, \dot{Y}, \dot{Z}]$ as the relative state vectors in the local-vertical-local-horizontal (LVLH) coordinate frame, the linearized dynamic equation of the relative motion can be described as:

$$\begin{bmatrix} \ddot{X} \\ \ddot{Y} \\ \ddot{Z} \end{bmatrix} = \begin{bmatrix} \omega^2 X - \dfrac{\mu}{R^3} X + 2\omega\dot{Z} + \dot{\omega} Z \\ -\dfrac{\mu}{R^3} Y \\ \omega^2 Z + 2\dfrac{\mu}{R^3} Z - 2\omega\dot{X} - \dot{\omega} X \end{bmatrix} + \mathbf{u} + \mathbf{d} \tag{3.1}$$

where $\mathbf{u} = [u_x, u_y, u_z]$ represents the actual control acceleration input vector applied to the chaser, μ is the Earth's gravitational constant and $\mathbf{d} = [d_x, d_y, d_z]$ is the external disturbance acceleration vector. Also, ω is the orbital rate of the rotating coordinate system, which is the time-derivative of true anomaly ν (i.e. $\omega = \dot{\nu}$) and can be derived from the following augmented equations [187]:

$$\dot{\omega} = -2\omega\frac{\dot{\mathbf{R}}}{\mathbf{R}} \tag{3.2}$$

$$\ddot{\mathbf{R}} = \mathbf{R}\omega^2 - \frac{\mu}{\mathbf{R}^2} \tag{3.3}$$

where $\mathbf{R}$ stands for the relative displacement vector of the target spacecraft from the center of the Earth. With $\mathbf{r}_i = [X_i, Y_i, Z_i]$ and $\mathbf{v}_i = [\dot{X}_i, \dot{Y}_i, \dot{Z}_i]$ as the initial states, the transfer trajectory can be integrated.

The mass of the chaser is decreased during the rendezvous, depending on the amount of thrust acted on the spacecraft. Having $\mathbf{T} = [T_x, T_y, T_z]$ as the thrust vector, the variation of the spacecraft mass can be formulated as:

$$\dot{m} = -\frac{T}{I_{sp}g_0} \tag{3.4}$$

where I_{sp} is the specific impulse of the propellant that is used, g_0 is the sea-level standard acceleration of gravity ($g_0 = 9.807m/s^2$) and T is the magnitude of the thrust vector. Consequently, $\mathbf{u} = \mathbf{T}/m$ gives the actual control acceleration input from the thrust vector. Considering t_f as the final rendezvous time, if the time profile of the thrust vector $\mathbf{T}$ is known, integrating the proposed system of equations in the time interval $0 < t < t_f$ yields the time histories of the relative position and velocity of the chaser spacecraft. The final states as $\mathbf{r}_f = [X_f, Y_f, Z_f]$ and $\mathbf{v}_f = [\dot{X}_f, \dot{Y}_f, \dot{Z}_f]$ are obtained at the end of the space rendezvous.

3.2.2 Continuous Thrust Modeling

Parameterizing the thrust vector has a significant impact on the convergence of the optimization algorithm in direct approaches. In this research, the magnitude of thrust in each direction T_x, T_y and T_z is approximated by considering N_p number of interpolation points in the desired time interval $0 < t < t_f$ in the allowable thrust limits of $T_{min} < [T_x, T_y, T_z] < T_{max}$. Having N_p number of uniformly discretized points, the overall time span is divided into $N_p - 1$ sub-intervals. The interpolating polynomial for the time interval can be represented by:

$$\hat{T}(t) = \sum_{k=1}^{N}(\prod_{j\neq k}\frac{t - t_j}{t_k - t_j})p_k \tag{3.5}$$

where $\hat{T}(t)$ denotes any of the thrust components (T_x, T_y, T_z), t_k is the discretized times, and p_k is the discrete points within the time interval. Given the number of discrete points N_p for each thrust component, the thrust profile may be interpolated with different shapes. One of the most popular methods is using piecewise cubic Hermite interpolating polynomials [188]. Various

types of splines can be obtained depending the choice of tangents in each node. Three practical types of splines from the family of Hermite splines, which are frequently used in many applications, are illustrated in Fig. 3.1 for approximating the thrust components.

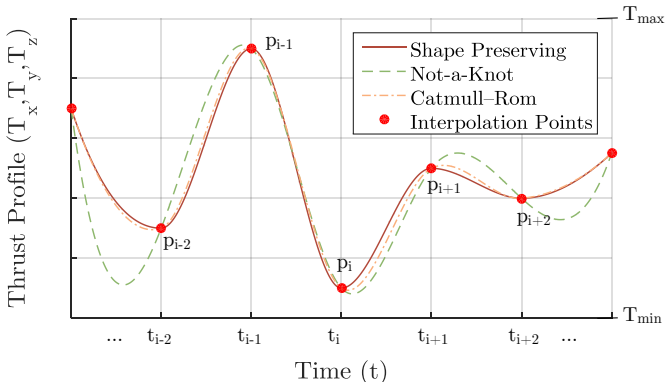

Fig. 3.1: Thrust profile interpolating with piecewise cubic Hermite splines.

These splines include Shape Preserving (SP) spline [189], Not-a-Knot (NK) spline [190], and Catmull-Rom (CR) spline [191]. All of these are continuous and have a continuous first derivative. The difference between these polynomials is the choice of tangents in the internal and end nodes. SP spline is designed so that it never locally overshoots the data. The slope at each interior point is taken to be a weighted harmonic mean of the slopes of the piecewise linear interpolant. One-sided slope conditions are imposed at the two end points. The slopes at the nodes can be computed without solving a linear system. NK spline is smoother spline and it also has continuous second derivatives. Even the third derivative is continuous up to some round-off error [192]. CR spline is yet another member of the Hermite family, and it has a balanced smoothness. The slope of the CR spline at data points depends directly on the points before and after. The resulting piecewise cubic does not have a continuous second derivative and it does not always preserve shape. However, it can be evaluated quickly by a convolution operation. More details regarding the derivation of each of these splines is beyond the scope of this chapter and the reader is urged to refer to the provided references for details [189, 190, 191].

3.3 Optimization Approach

Having the orbital elements of the target space vehicle and initial relative position and velocity of the chaser, the initial state values of Eq. 3.1 becomes

known. Also, knowing the initial mass of the chaser, along with the specific impulse of the propellant, makes it possible to propagate the transfer trajectory if the thrust profile is known. Aiming at the short-range space rendezvous, it is now possible to formalize a continuous optimization problem. Particularly, for a predefined short-range rendezvous mission, the main goal is to find the optimal variation of thrust components in each axis, so that the chaser reaches the target spacecraft with minimum fuel consumption. While the fuel mass is considered as the objective function, the final relative distance and final relative velocity of the chaser are the main constraints in the problem. Regarding this explanation, short-range space rendezvous can be defined as a constrained optimization problem with inequality constraints. The general form of such a constrained optimization problem is:

$$\begin{aligned} &Minimize && F(x) \quad x = (x_1, x_2, ..., x_n) \\ &Subject\ to && G_i(x) \leq 0 \quad i = 1, ..., n_g \\ & && H_i(x) = 0 \quad i = 1, ..., n_h \\ & && x_{min} < x_i < x_{max} \end{aligned} \tag{3.6}$$

where the goal is to minimize the objective function $F(x)$ with respect to n dimensional parameter vector $x \in R^n$, while the feasible region is restricted by x_{min} and x_{max} as the lower and upper boundaries, $G_i(x)$ as the function for n_g inequality constraints and $H_i(x)$ as the function for n_h equality constraints. Based on the proposed formulation of the short-range space rendezvous problem, the decision variables $x = (x_1, x_2, ..., x_n)$ are the interpolation points for three components of thrust vector:

$$x = p_i \quad (i = 1, ..., 3N_p) \tag{3.7}$$

where $n = 3N_p$, and the boundaries are $x_{min} = T_{min}$ and $x_{max} = T_{max}$. Also, the objective function is defined as:

$$F(x) = m_f = \int_0^{t_f} \dot{m} dt \tag{3.8}$$

where m_f is the fuel mass, consumed by the chaser spacecraft within the transfer. Regarding the constraints, the distance and velocity of the chaser relative to the target at the end of the transfer are represented as:

$$G(x) = \begin{bmatrix} |\mathbf{r}_f| - \sigma_r \\ |\mathbf{v}_f| - \sigma_v \end{bmatrix} \tag{3.9}$$

where σ_r and σ_v are the expected final distance and velocity of the chaser relative to the target spacecraft, and these are specified as the space mission requirement. In the trajectory optimization of spacecraft short-range rendezvous in this research, the employed approach does not force any equality

constraints. However, the proposed algorithm is developed for facing the constrained optimization problems in general form, i.e., dealing with all types of constraints.

To deal with this constrained continuous optimization problem, an EDA-based algorithm is proposed in the following section. In this research, the main contribution is to present a new concept toward EDAs for handling constraints. In this regard, two of the mechanisms in EDAs including **SEEDING** and **LEARNING** have been enhanced and one additional mechanism, named **MAPPING**, have been developed in this research. Having these new mechanisms, the optimization process is modified towards satisfying the constraints, while minimizing the objective function. It is noteworthy that in the developed algorithm, if an initial feasible population is provided either directly or as the output of the seeding process, the feasibility of the final solution is guaranteed. Simplicity and robustness relative to the type of the problem and constraints are two main aims in developing this method.

EDAs are a type of population-based evolutionary algorithms designed for solving numerical optimization problems. Based on machine learning techniques, at each iteration, EDAs learn a probabilistic model from a subset of the most promising solutions, trying to explicitly express the interrelations between the different variables of the problem. Then, by sampling the probabilistic model learned in the previous iteration, a new population of solutions is created. In other words, EDAs work based on two major key methods: learning and sampling, where a probabilistic model that estimates the probability distribution of the selected solutions is learned and then utilized for sampling new individuals [180]. However, in constrained continuous optimization, there are no guarantees that the newly obtained solutions satisfy the constraints of the problem. As previously mentioned, the mechanisms proposed in this work are introduced in the framework of EDAs, named EDA++. The overall pseudocode of the proposed algorithm is in Algorithm 2.

Following the pseudocode of Algorithm 2, the overall optimization process is as follows. In EDA++, the optimization process starts by forming the function $C(x)$ for measuring the infeasibility of solutions as the constraint violation:

$$C(x) = \frac{\sum_{i=1}^{n_g} \hat{G}_i(x) + \sum_{j=1}^{n_h} \hat{H}_j(x)}{n_g + n_h} \tag{3.10}$$

with $\hat{G}_i(x)$ and $\hat{H}_j(x)$ defined by:

$$\hat{G}_i(x) = \max(0, G_i(x)) \tag{3.11}$$

$$\hat{H}_j(x) = \begin{cases} |H_j(x)| & |H_j(x)| > \epsilon \\ 0 & |H_j(x)| \leq \epsilon \end{cases} \tag{3.12}$$

Algorithm 2: Overall pseudocode of EDA++

Input: $F(x), G_i(x), H_i(x), x_{min}, x_{max}$
Parameters: $N, M, \epsilon, \gamma, S, \tau, \alpha, \lambda, N_\delta$, MapType
CONSTRUCT $C(x)$ FROM $[G_i(x), H_i(x), \epsilon]$
$[x,c,i] \leftarrow$ **SEEDING**$(C(x),x_{min},x_{max},N,M,\gamma,S,\tau)$
$f \leftarrow$ **EVALUATION**$(x,F(x))$
if $i < M$ **then**
  for $iter \leftarrow i$ **to** M **do**
    $[x_{sel}, f_{sel}] \leftarrow$ **SELECTION**(x,f,γ)
    $[\Phi, \phi] \leftarrow$ **LEARNING**$(x_{sel},f_{sel},C(x),\ \alpha,\lambda)$
    $x_{sam} \leftarrow$ **SAMPLING**(Φ,ϕ,N)
    $x_{rep} \leftarrow$ **REPAIRING**$(x_{sam},x_{min},x_{max})$
    $x_{map} \leftarrow$ **MAPPING**$(x_{rep},\Phi,\phi,\ C(x),N_\delta,$MapType$)$
    $f_{map} \leftarrow$ **EVALUATION**$(x_{map},F(x))$
    $[x, f] \leftarrow$ **REPLACEMENT**(x_{map},f_{map},x,f)
    EXTRACT $[x_{best}, f_{best}]$ FROM $[x, f]$;
    if *stopping criteria are met* **then**
      BREAK;
    end if
  end for
else
  EXTRACT $[x_{best}, f_{best}]$ FROM $[x, f]$;
end if
Output: x_{best}, f_{best}

where ϵ is the error margin for equality constraints. Having N as the population size and M as the maximum number of iterations, the **SEEDING** mechanism is utilized to generate an initial feasible population. Having the initial feasible solutions, with corresponding objective values f obtained from **EVALUATION**, the main optimization loop starts. At each iteration, the algorithm begins by selecting the top promising individuals in the current population according to the **SELECTION** method. Truncation selection method [180] is used in this research, with γ as the truncation factor. Having the selected population x_{sel} and the corresponding objective values f_{sel}, a probability model is learned via the **LEARNING** mechanism. In the proposed learning mechanism, the selected population is divided into several clusters of solutions (Φ and ϕ) with respect to their constraint violation. Then, a mixture of models is learned, one component on top of each cluster, in such a way that the probability of sampling feasible solutions becomes high. Having the mixture of model, new solutions are sampled via the **SAMPLING** method as x_{sam}. The **REPAIRING** method, simply refines the newly sampled solutions based on the boundaries of the solution domain x_{min} and x_{max}. Up to this point, the obtained solutions x_{rep} are likely to be inside the feasible region

thanks to the seeding mechanism and the proposed learning mechanism that is intended to generate feasible solutions. However, even doing that, the algorithm sometimes generates infeasible solutions. As a result, a **MAPPING** mechanism guarantees the feasibility and maps all possible infeasible solutions to the feasible region to form a completely feasible population x_{map}. After evaluating the objective value of the obtained solutions f_{map} via the **EVALUATION** process, the new individuals are combined with the individuals from the previous population, and the **REPLACEMENT** mechanism is invoked to form the new population and the corresponding objective values f in the current iteration. Population aggregation method is used for this mechanism in this research.

The mentioned process continues until at least one stopping criteria are met. As explained, EDA++ benefits from three newly developed mechanisms, which are distinct from the typical EDAs. These mechanisms, including seeding, learning and mapping, are described in details in the following sections.

3.4 Seeding

Providing initial feasible solutions is a priority in EDA++. The aim of the seeding mechanism is to ensure that the initial population is feasible regardless of the objective value of the solutions. The initial population containing only feasible solutions may be available and seeded to the algorithm initially. In this case, the seeding mechanism is skipped. However, if no initial feasible population is provided, the seeding mechanism is invoked. The pseudocode of this mechanism is shown in Algorithm 3.

As it is shown, the seeding mechanism includes an iterative optimization process based on a multivariate Gaussian EDA that considers the constraint violation function $C(x)$ in Eq. 3.10 as the temporary objective function. In this process, first an initial random population is created based on a uniform distribution of solutions within the boundaries of x_{min} and x_{max}. Then, the amount of constraint violation of the population is evaluated and if any infeasible solution exist within the population, the mechanism performs the multivariate Gaussian EDA to minimize the constraint violation. This EDA simply includes the sequence of learning, sampling, repairing, evaluation, and replacement. In this research, the learning and sampling stage within the seeding mechanism is based on multivariate Gaussian distribution. However, it can be other EDAs as well, including the one with the advanced learning mechanism, which is described in the next section. This iterative process stops when all of the solutions in the population are feasible. Also, this process restarts every S number of iterations, while saving the high quality solutions in terms of constraint violation. In case of a restart, the top τ fraction of the solutions with lowest constraint violation is saved and added to a new population of solutions with random uniform distribution. The goal of this restart procedure is to speed up the process of reaching feasible solutions, while avoiding the

Algorithm 3: Pseudocode of the seeding mechanism

Input: $C(x)$,x_{min},x_{max},N,M,γ,S,τ

$retryFlag \leftarrow false$
$i \leftarrow 0$
while $i < M$ **do**
  $i \leftarrow i + 1$
  if $i = 1$ **then**
    $x \leftarrow$ UNIF. DIST. $[x_{min},x_{max},N]$
    $c \leftarrow$ **EVALUATION**$(x,C(x))$
  else if $retryFlag = true$ **then**
    $retryFlag \leftarrow false$
    $x_1 \leftarrow$ **SELECTION**(x,c,τ)
    $x_2 \leftarrow$ UNIF. DIST. $[x_{min},x_{max},N(1-\tau)]$
    $x \leftarrow [x_1, x_2]$
    $c \leftarrow$ **EVALUATION**$(x,C(x))$
  else
    $[x_{sel}, c_{sel}] \leftarrow$ **SELECTION**(x,c,γ)
    $\Phi \leftarrow$ **LEARNING**(x_{sel},f_{sel})
    $x_{sam} \leftarrow$ **SAMPLING**(Φ,N)
    $x_{rep} \leftarrow$ **REPAIRING**$(x_{sam},x_{min},x_{max})$
    $c_{rep} \leftarrow$ **EVALUATION**$(x_{rep},C(x))$
    $[x, c] \leftarrow$ **REPLACEMENT**(x_{rep},f_{rep},x,c)
  end if
  if $max(c) = 0$ **then**
    BREAK;
  else if $min(c) = 0$ **then**
    CONTINUE;
  else if $REMINDER(i, S) = 0$ **then**
    $retryFlag \leftarrow true$;
end while

Output: x, c, i

seeding mechanism to get trapped in a local optima. Note that the seeding mechanism does not trigger the restart flag if at least one feasible solutions is found. The seeding mechanism stops once the sufficient number of feasible solutions are achieved, or the mechanism reaches the maximum number of iterations. After the seeding mechanism, the objective values of the obtained solutions are evaluated.

3.5 Learning

Having a feasible population, obtained from the seeding mechanism, the main loop of the optimization starts. A selection of high quality feasible solutions x_{sel} along with their corresponding objective values f_{sel} is chosen from the current population, and these are used to estimate the parameters of the probability model. The pseudocode of the learning mechanism is shown in Algorithm 4.

Tha main idea of the learning process is based on utilizing a mixture of Gaussian distributions as probabilistic model whose density function is formalized as:

$$f(x) = \sum_{k=1}^{N} \pi_k f_k(x|\mu_k, \Sigma_k) \tag{3.13}$$

where each $f_k(x|\mu_k, \Sigma_k)$ component of the mixture is a multivariate Gaussian distribution and μ_k and Σ_k are the mean value (the centroid) and the covariance matrix of the k model for $k = 1, \ldots, N$ respectively with π_k as the mixing coefficient for the kth component.

Algorithm 4: Pseudocode of the learning mechanism

Input: x_{sel},f_{sel},$C(x)$, α,λ
$N_{sel} \leftarrow size(x_{sel})$
for $i \leftarrow 1$ **to** N_{sel} **do**
  $[\iota, \mu] \leftarrow$ kmeans(x_{sel},i);
  $c_\mu \leftarrow$ **EVALUATION**(μ,$C(x)$)
  if $max(c_\mu) = 0$ **then**
    BREAK;
end for
CONSTRUCT Φ FROM $[\mu, x_{sel}(\iota)]$
$N_c \leftarrow size(\Phi)$
for $i \leftarrow 1$ **to** N_c **do**
  EXTRACT $[\hat{x}, \hat{f}, \hat{\mu}, \hat{\sigma}]$ FROM $\Phi(i)$
  $[\hat{x}_{sel}, \hat{f}_{sel}] \leftarrow$ **SELECTION**($\hat{x}$,$\hat{f}$,α)
  $d \leftarrow ||\hat{x}_{sel} - \hat{\mu}||$
  $j \leftarrow 0$
  if $d > \lambda \times \hat{\sigma}$ **then**
    $j \leftarrow j + 1$
    CONSTRUCT $\hat{\phi}$ FROM $[\hat{\mu}, \hat{x}_{sel}]$
    $\phi(j) \leftarrow \hat{\phi}$
end for
Output: Φ, ϕ

In the proposed learning stage, the Gaussian mixture model is constructed in two steps. The first step consists of finding the minimum number of mixture components in which all the centroids (μ_k) are placed inside the feasible region. To this end, an iterative clustering process is developed. The scheme of this process is illustrated in a schematic instance in Fig. 3.2.

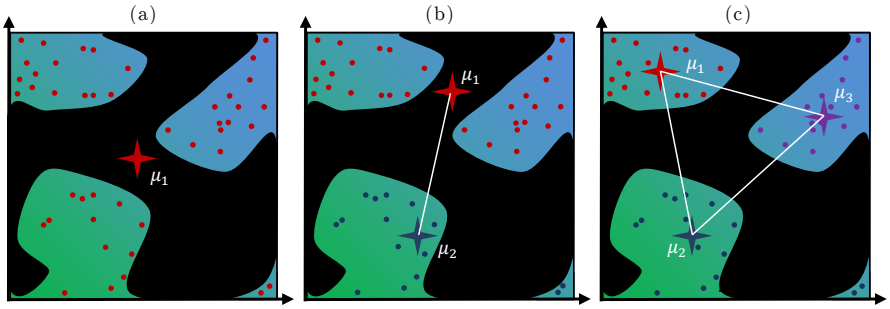

Fig. 3.2: Clustering the selected population within the learning mechanism

In the plots, the infeasible region due to the constraints is depicted in black, while the feasible region is illustrated as the color-mapped area. The selected population is plotted and different numbers of clusters (k) are considered. In this research, k–$means$++ is chosen as the clustering method [193]. However, other methods could also be considered. In plot (a), just one cluster is considered ($k = 1$) and thus there is one centroid, which is the mean value of the population. As can be seen, in this case the centroid resides in the infeasible solution, and therefore, this probabilistic model does not meet the requirement of satisfying the constraints as the sampled solutions will be mostly in the infeasible region. In the plot (b), the population is divided into two clusters ($k = 2$). The positions of the centroids indicate that one of them is inside the feasible region, while the other one is not, leading to the conclusion that this mixture model is also not suitable for constraints satisfaction. Considering three clusters ($k = 3$), yields the plot (c) in Fig. 3.2. As can be appreciated, all centroids are inside the feasible region. Therefore, the mixture of Gaussian distributions model is learned by calculating the maximum likelihood estimators of the parameters of the components in this mixture, using the solutions in the respective clusters. Since all of the centroids are feasible in this case, any solution that is going to be sampled is likely to be inside the feasible domain.

This process is the first loop in Algorithm 4, representing the first step of the learning process. The obtained number of clusters N_c in this scenario is the minimum number of clusters with feasible centroids. Finalizing the process, the components Φ, referred to as the *parent clusters* are extracted, which contain corresponding solutions $\hat{x}$, objective values $\hat{f}$, centroids $\hat{\mu}$ and

covariances $\hat{\sigma}$. Although, it is possible to continue increasing the number of components and obtain other mixtures of Gaussian distributions, the computation time will increase without any actual necessity as the objective is to find a minimum number of mixture components with feasible centroids. The main benefit of such process is that having all of the centroids ($\hat{\mu}$) from the components inside the feasible region significantly reduces the chance of sampling infeasible solutions later on during the sampling process.

The described learning process satisfies the primary requirement for sampling feasible solutions. However, when the mapping mechanism (explained in the next section) is applied to the model that has been created based on this learning process, the covariance matrix tends to shrink, i.e. looses diversity. This effect reduces the convergence rate of the optimization process. To overcome this drawback, in the next step of the learning process, more components are added to the model. This step is to compensate the covariance loss due to the mapping mechanism that is going to be used in the algorithm [194] after the sampling stage. In this step, for each component Φ_i, first, the top α percentage of the best solutions ($\hat{x}_{sel}$ and $\hat{f}_{sel}$) are selected. Then, the selected set of solutions is analyzed to see if they have outliers using the Z-score outlier detection method [195] with respect to a given distance threshold λ from their respective centroids $\hat{\mu}$. According to this mechanism, if an outlier solution is at the top α percentage of the best solutions, it will be considered as the centroid of a new component in the mixture $\hat{\phi}$, referred to as a *outlier-based cluster*. For the newly formed components, we assume an independent multivariate Gaussian distribution, where the variance of each dimension is calculated as the half of the distance from the initial centroid in each component. The illustration of this approach in a schematic instance is shown in Fig. 3.3.

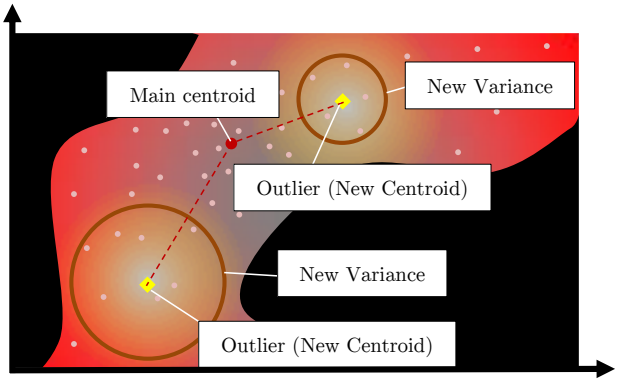

Fig. 3.3: Formation of outlier-based clusters within the learning mechanism

As shown in this instance, the selected population inside the feasible region is depicted along with the corresponding centroid. According to these parameters, two outliers are detected that have the objective function values above a predefined threshold in this population, and they are therefore considered as the centroids for two new components in the mixture. The variance of the independent multivariate Gaussian distributions for each newly generated component is considered as half of the distance from the outlier to the centroid in the component. Overall, a mixture model of three Gaussian models is determined in this iteration: one initial component (parent cluster) and two additional components (outlier-based clusters) due to outliers.

3.6 Mapping

Although the proposed learning mechanism generates mixture components with mean values inside the feasible region, when sampling, it is still possible that some samples are generated outside the feasible region. In order to solve this problem, a mapping mechanism is introduced, which is utilized after the repairing process. The application of this mapping mechanism in a schematic instance is illustrated in Fig. 3.4.

The presented plots illustrate the mapping mechanism in one iteration of the optimization process. As it is shown, the mapping method is based on the idea of shifting infeasible points toward their respective centroid in N_δ number of steps until they enter the feasible region. As shown in Fig. 3.4 (a), the probabilistic model in this iteration is a mixture of two Gaussian models. New solutions are sampled around the centroids. However, not all the samples are inside the feasible region. The feasible and infeasible solutions are marked separately, connected to their respective centroid. The points are mapped in a deterministic equally-spaced form toward the centroids with N_δ as the maximum number of steps. The amount of displacement in each step is $(\mu_i - x_j)/N_\delta$, where x_j is the infeasible solution to be mapped toward the centroid μ_i in the ith cluster. As a result, the distance between the infeasible point and the respective centroid is divided into N_δ steps. It is worth noting that, at the final step, the last displacement will be on the centroid. So, the feasibility is guaranteed no matter what the number of N_δ is. However, the higher the number of N_δ is, the more accurate the feasible and infeasible borders that will be discovered. Note that while moving toward the centroids, the mapping is stopped as the point enters the feasible region. The mapped solutions are shown in Fig. 3.4 (b).

Regarding the mapping mechanism, one can consider different approaches depending on their non-linear or stochastic nature. In the following, four alternatives are proposed.

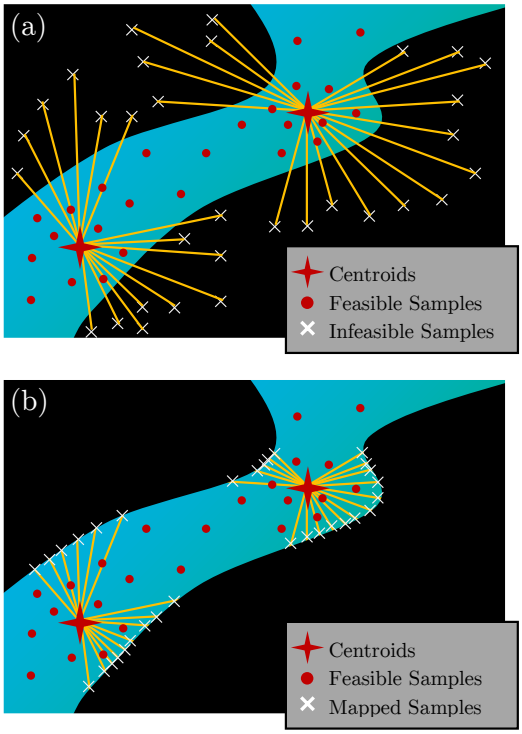

Fig. 3.4: The process of shifting infeasible individuals toward the centroids, (a) Before mapping (b) After mapping.

3.6.1 Linear Deterministic Mapping

The linear deterministic (LD) mapping is a straight forward method for shifting the infeasible point toward the respective centroid. In this method, the distance between the infeasible solution and the centroid is linearly divided into equal steps. The infeasible solution is moved from its initial position toward the centroid with respect to the steps. In each step the feasibility of the new solution is checked. The process stops when the shifted solution has entered the feasible region.

This mapping process can be represented as:

$$x_j^{new} = x_j + \delta \tag{3.14}$$

where x_j is the current point within the infeasible region, x_j^{new} is the new shifted solution towards the respective centroid μ_i in the component, and δ is the step calculated as:

$$\delta = \frac{|\mu_i - x_j|}{N_\delta} \tag{3.15}$$

where N_δ is the selective total number of steps for this mapping mechanism.

3.6.2 Linear Stochastic Mapping

The linear stochastic (LS) mapping is similar to LD. The only difference is that in each step, after obtaining the new solution, it will also be shifted in a random direction with a variable radius r as:

$$x_j^{new} = x_j + r \times \delta \tag{3.16}$$

where $0 < r < 1$. This forces a random movement of the point while mapping and may have some advantages depending on the solution domain as it produces diversity to the search.

3.6.3 Bisection Deterministic Mapping

Bisection deterministic (BD) mapping is based on repeatedly bisecting the interval defined by the centroid and the infeasible solution. At each step, the distance is divided in two by computing the midpoint of the interval as:

$$\delta = \frac{|\mu_i - x_j|}{2} \tag{3.17}$$

The feasibility of the midpoint solution is evaluated. If the new solution is feasible, the process stops. Otherwise, the process continues considering the interval between the new obtained solution and the centroid. This process is similar to the well-known bisection method in finding the root of a continuous function.

3.6.4 Bisection Stochastic Mapping

Likewise, the bisection stochastic (BS) mapping is similar to BD. The difference is that in each step, after obtaining the new solution, it will also be shifted in a random direction with a variable radius $0 < r < 1$.

3.7 Algorithm Verification

In order to analyze the performance of the proposed EDA++, the experiments have been carried out on different benchmarks of problems. The algorithm performance is compared with state-of-the-art algorithms and constraint handling methods[1]. First, the efficiency of the algorithm is tested against other constraint handling techniques on the well-known benchmark suite [196], which

[1] All materials for the experiments, including the codes and the results are available at https://github.com/abolfazlshirazi/EDAPP

contains 13 constrained optimization problems, and also the proposed mapping mechanisms are analyzed. In the second experiment, the performance of the algorithm is compared with state-of-the-art algorithms on CEC 2020 test-suite benchmark [197], which contains 57 non-convex constrained optimization problems.

3.7.1 Common parameters setup

The experiments are conducted on HIPATIA cluster setup of BCAM, with 18 nodes including 672 cores (Processor Intel(R) Xeon(R) Gold 6140 CPU @ 2.30GHz) and 3360 GB RAM for the aforementioned runs. In all of the experiments, the following predefined parameters are chosen for EDA++ since the main goal of the experiments is to compare the performance of EDA++ without tuning the parameters. The initial parameters of N, M, and ϵ are set according to each benchmark. In the seeding mechanism, parameters S and τ are considered as 100 and 0.2. The truncation factor γ for the selection method is chosen as 0.5. Within the learning process, the multivariate Gaussian model is utilized as the component mixture and k–$means$++ [193] is used as the clustering method. The outlier detection parameters are chosen as $\lambda = 1$ and $\alpha = 0.01$. Within the sampling process, for a new size N population, several choices exist for the number of samples for each mixture component. The typical option, which is used in this research, is to dedicate an equal sample size to each component (i.e., N/k) for k components. However, this is an optional choice. Obviously dedicating more samples to the parent clusters or outlier-based clusters acts as a balancing parameter for exploration/exploitation behavior of the optimization algorithm. Also, the number of steps for the mapping mechanism is chosen as $N_\delta = 10$ for all proposed mapping methods. The type of the employed mapping mechanism will be specified in each experiment.

3.7.2 Analysis of EDA++ components

In the first experiment, the efficiency of the proposed algorithm is analyzed with respect to other constraint handling techniques, combined with well-known EAs. The benchmark suit consists of 13 constrained optimization problems in the well-known benchmark of [196] is considered in this experiment. In this regard, the performance of the proposed EDA++ is compared with three algorithms: Genetic Algorithms (GA), Particle Swarm Optimization (PSO) and Covariance Matrix Adaptation Evolutionary Strategy (CMA-ES) [198]. As for the GA, the crossover rate, crossover range factor, mutation percentage and mutation rate of 0.7, 0.4, 0.3 and 0.2 are chosen, respectively. Also, the PSO is a vectorized Particle Swarm Optimization with personal learning coefficient and global learning coefficients equal to 2, and inertia weight damping ratio of 0.99. For CMA-ES, default values are used for its parameters as described in [198]. Since the best parameter selection for the algorithms depends on each specific problem, these values are chosen arbitrary for each algorithm.

RBP							
#	GA	PSO	CMA-ES	EDA++ (LD)	EDA++ (LS)	EDA++ (BD)	EDA++ (BS)
1	4.01e-09	0	1.51	2.73e-07	2.34e-06	3.64e-06	5.42e-06
2†	2.5	42.5	50.1	5.59	0	21.6	9.4
3†*	27.4	0.476	91.9	0.00177	0.00187	0.00269	0
4	0.161	4.31e-06	0.229	1.48e-08	0	6.19e-08	1.6e-08
5*	0.000272	0.000272	0.000717	0	2.71e-07	5.07e-07	1e-06
6	1.04	1.48	1.27	0.00016	0.000133	0	5.7e-05
7	0.871	0	17.3	1.47	0.104	2.38	1.66
8†	1.45e-14	3.48e-07	0	1.45e-14	1.45e-14	1.45e-14	1.45e-14
9	0.00707	0.00402	0.761	0.00387	0	0.00724	0.000272
10*	2.78	0	84.2	3.06	2.8	7.81	4.2
11	0.0237	0.00323	0.151	2.6e-06	2.26e-05	0	4.24e-07
12†	0	2.91e-11	6.82e-07	0	0	0	0
13*	0.202	0.202	228	0.0446	0.046	0	0.0463
ARPD							
#	GA	PSO	CMA-ES	EDA++ (LD)	EDA++ (LS)	EDA++ (BD)	EDA++ (BS)
1	5.82e-09	7.49	3	3.48e-05	1.33	1.33	1.33
2†	5.44	52.4	63.1	17.9	15.1	41	27.3
3†*	51.5	1.52	98.3	1.06	0.274	0.0199	0.00575
4	0.436	3.21e-05	1.91	3.23e-05	5.87e-06	1.9e-05	2.63e-05
5*	0.000715	0.000715	4.44	0.000466	0.000446	0.000439	0.000425
6	9.4	2.88	6.26	0.00209	0.00188	1.07e-06	0.000209
7	2.63	2.22	545	5.5	2.76	6.16	7.76
8†	3.77e-14	3.91e-05	0.085	2.03e-14	1.74e-14	2.03e-14	2.17e-14
9	0.043	0.0138	18.9	0.0172	0.00802	0.0246	0.00606
10*	8.24	2.69	193	16.7	13.7	20.5	16.5
11	0.101	0.0643	2.58	0.00113	0.00133	0.000164	0.000248
12†	0	4.54e-10	0.00414	0	0.0563	0.0563	0.0563
13*	10.8	10.8	5.7e+03	10.6	10.5	10.4	10.2

Table 3.1: RBP and ARPD values of the feasible solutions obtained after 10 runs.
$*$ Equality constraints are converted into inequality constraints with $\epsilon = 10^{-3}$.
† Maximization problems are converted into minimization problems.

Different constraint handling methods are employed in GA, PSO and CMA-ES. Handling the constraints in GA and PSO is based on a static penalty function with a constant coefficient. Several penalty factors, such as $1, 100$ and 10000, are tested and the best performance of GA and PSO was found to be that with the highest value. In CMA-ES, the resampling technique, as used in [199], was considered as the constraint handling technique and the best obtained feasible solution is saved in each iteration.

In the first experiment, each algorithm is run 10 times for each of the 13 problems of the benchmark. All algorithms started with feasible initial populations. Also, the same initial population is considered in the run of each algorithm in order to have a fair comparison. Considering D as the number of dimensions in each problem, the population size and the number of iterations are considered as $20 \times D$ and $30 \times D$ respectively and no additional stopping criterion is assumed.

A summary of the obtained results are tabulated in Table 3.1. In this table, the four proposed mapping mechanisms are tested in EDA++ along with GA, PSO and CMA-ES. The Relative Best Percentage (RBP) and Average Relative Percentage Deviation (ARPD) values [182] of the obtained feasible solutions across 10 repetitions by each algorithm are provided. The results for PSO and GA are regarding the penalty factor of 10000 and as mentioned, CMA-ES benefits from the resampling technique for handling constraints. According to the obtained results, although none of the algorithms could find the global optimal solution in all problems, the performance of the proposed algorithm is competitive against GA and PSO incorporated with penalty function and CMA-ES with resampling technique. Comparing the best and mean values of the final solutions confirms the high efficiency of EDA++ relative to others. It is worth noting that unlike GA, PSO and CMA-ES, EDA++ always returns feasible solutions. In some cases, GA or/and PSO failed to reach feasible solutions, even when the provided initial population is feasible. These cases include 1 run of GA for problem *g10*, 2 runs of PSO for problem *g06* and 4 runs of PSO for problem *g10*. On the other hand, the feasibility of the final solution is guaranteed in EDA++, when the initial feasible solution is provided.

CMA-ES does not return infeasible solutions as it is associated with a trigger that returns the best feasible solution found so far in every iteration after the resampling process. Evaluating the results for CMA-ES shows that despite returning feasible solutions in all cases, solutions by CMA-ES have the lowest quality in comparison to GA, PSO and EDA++. Considering the quality of the obtained solutions in Table 3.1, EDA++ is either quite superior or has the same performance as the other algorithms in finding good feasible solutions.

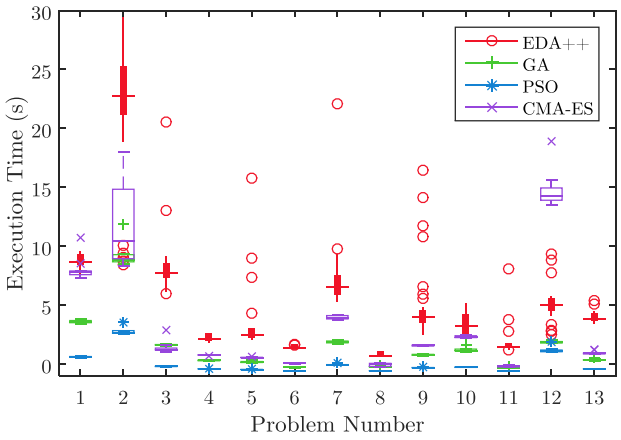

Fig. 3.5: Comparison of the execution times of the algorithms

Analysis of the time burden of the proposed algorithm shows how competitive the presented EDA++ is in comparison to other algorithms. This analysis is depicted in Fig. 3.5. In this figure, the boxplot for the execution time of all algorithms is plotted for each problem. All runs are considered in this plot, regardless of whether they achieve feasible or infeasible solutions. Results indicate that the vectorized PSO has the fastest process, due to the parallel computing associated with the structure of the code. On the other hand, the execution time of the proposed algorithm is competitive with GA and CMA-ES. Also, it can be seen that the variance of execution time is higher in EDA++. This is due to the fact that various mapping mechanisms perform several iterations per each infeasible individual in every generation. Depending on how the infeasible solutions are distributed, the mapping mechanism takes variable times. Therefore the number of iterations for mapping infeasible solutions toward the respective centroids, makes the algorithm take longer to converge. However, in an exchange for having all feasible solutions by the proposed EDA, the time burden is acceptable and competitive.

Detailed comparison between the proposed mapping mechanisms is depicted in Fig. 3.6. In this figure, two kinds of plots are illustrated for each of the problems. The top graphs are dedicated to the boxplots for the quality of the obtained solution and the second graphs include the boxplots for the execution times of the algorithms. Each boxplot separates the data for all four aforementioned mapping mechanisms. Results shows that none of the mapping mechanisms outperforms the rest in all problems. Therefore, the efficient mapping mechanism depends on the type of the constrained optimization problem.

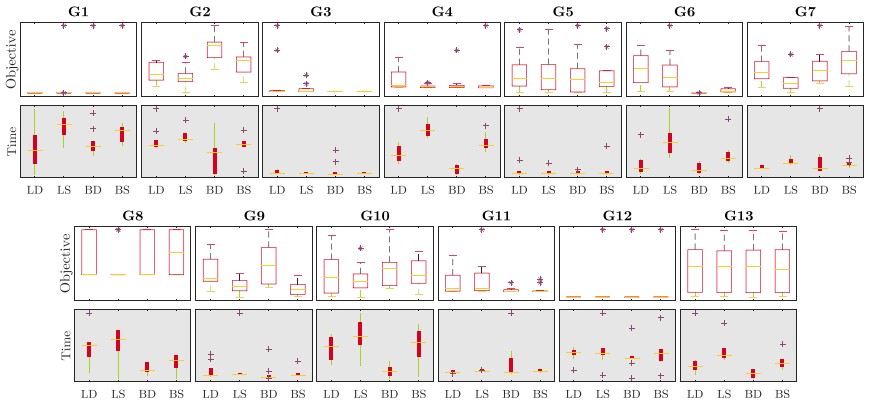

Fig. 3.6: Comparison of mapping mechanisms

3.7.3 Comparison with the state-of-art algorithms

In this experiment, we focused on the state-of-the-art algorithms in CEC 2020 competition for real-world single objective constrained optimization and the corresponding benchmark problems [197]. An overview of the problems' features in this benchmark is shown in Fig. 3.7. This figure shows the dimension, number of inequality and equality constraints and the feasibility ratio (F.R.) percentage [1] of the solution domain for all 57 problems. As it is shown, the majority of the problems have zero F.R. percentage due to the existence of equality constraints. Even some of the problems with only inequality constraints have zero F.R. percentage, since the defined inequality constraints have the effect similar to equality constraints. Also, two problems have F.R. percentage of 100%, which makes them almost unconstrained problems.

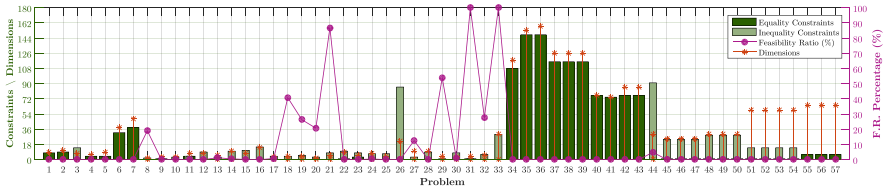

Fig. 3.7: Features of the benchmark problems in CEC 2020 competition on real-world constrained optimization

Some of the state-of-the-art algorithms in this competition were chosen initially as candidate algorithms for comparison[2]. The algorithms include SASS [200], COLSHADE [201], ϵsCMAgES [202], EnMODE [203], and BP-ϵMAgES [204]. Although these algorithms showed quite good performance in the competition, detailed analysis indicates that some of the them take advantage of specific features, which makes the comparison not quite reasonable and fair. Some remarks regarding the comparison worth to note here. For example, SASS and ϵsCMAgES use analytical Jacobian, while satisfying the constraints and minimizing the objective function. It means the algorithms will not work if the Jacobian matrix is not provided. On the other hand, calculating the Jacobian matrix numerically takes enormous number of function evaluations. In other words, analytical derivation of Jacobian prior to the optimization saves thousands of function evaluations, which makes the comparison unfair since the rest of the algorithms do not take advantage of analytical Jacobian. The other issue for SASS and ϵsCMAgES is that they use

[1] This feature is measured by evaluating 10000 random inputs, with uniform distribution between the upper and lower bounds for each problem.

[2] Please refer to the CEC 2020 RW Constrained Optimization repository for the available codes: https://github.com/P-N-Suganthan/2020-RW-Constrained-Optimisation

MATLAB optimization toolbox functions internally. In other words, there are no actual mechanisms to deal with constraints in these algorithms. The other issues, not only with these two algorithms, but also in some other algorithms in this competition is exceeding the function evaluation limits and tuning the parameters[1].

Having these considerations into account, COLSHADE, EnMODE, and BP-ϵMAg-ES are chosen for this experiment. LSHADE44 [205] from CEC 2017 competition is also considered as a complementary algorithm. The comparison setup is implemented same as CEC 2020 competition. Regarding this, the maximum allowable function evaluation for each problem is considered as:

$$MAX_{FEs} = \begin{cases} 1 \times 10^5 & D \leq 10 \\ 2 \times 10^5 & 10 < D \leq 30 \\ 4 \times 10^5 & 30 < D \leq 50 \\ 8 \times 10^5 & 50 < D \leq 150 \\ 10^6 & 150 < D \end{cases} \tag{3.18}$$

where D is the dimension of the problem. LD mapping method is chosen in this experiment and equality constraints are converted into inequality constraints with the threshold of $\epsilon = 10^{-4}$. The population size N for EDA++ is considered as

$$N = \min(200, \max(10 \times D, 50)) \tag{3.19}$$

Also, the default values of the parameters are considered for the rest of the algorithms. According to this setup, each algorithm is run 25 times for each problem and all of the solutions are saved. Results are provided in Table 3.2, including the percentage of feasibility rate (FR) and the mean value of constraint violations (MV). As it is shown, EDA++ managed to find at least one feasible solution (non-zero FR) similar to COLSHADE, LSHADE44 and EnMODE. In this regard BP-ϵMAg-ES has the best performance. However, for the majority of the problems (1 to 33) EDA++ outperforms BP-ϵMAg-ES in terms of FR percentage. Moreover, for the rest of the problems, EDA++ has competitive performance with COLSHADE, LSHADE44 and EnMODE. Same results can be observed regarding the MV values. Superior performance has been observed for EDA++ in most of the problems (2-6, 8-10, 12-15, 17-24, 27-33, 44, 45). The performance of COLSHADE and EnMODE are competitive in this matter and they are close to EDA++. For the other problems, one observes reasonable performance for EDA++ in lowering the constraint violations.

Besides the FR and MV values, the execution time of the algorithms are compared in Fig. 3.8. As the number of function evaluations are the same for each problem, the relative difference in the execution times can be analyzed.

[1] Detailed discussions are omitted here due to brevity, and the reader is urged to refer to the provided codes in the CEC 2020 repository.

#	FR					MV				
	EDA++	BP-ϵMAg-ES	COLSHADE	LSHADE44	EnMODE	EDA++	BP-ϵMAg-ES	COLSHADE	LSHADE44	EnMODE
1	28	80	0	4	100	2.0e-03	1.0e+01	1.9e-01	3.1e-03	0.0e+00
2	100	44	100	100	100	0.0e+00	2.3e+01	0.0e+00	0.0e+00	0.0e+00
3	100	64	100	100	100	0.0e+00	8.9e+02	0.0e+00	0.0e+00	0.0e+00
4	100	100	100	100	100	0.0e+00	0.0e+00	0.0e+00	0.0e+00	0.0e+00
5	100	52	100	100	100	0.0e+00	1.8e+01	0.0e+00	0.0e+00	0.0e+00
6	0	20	0	0	0	5.6e-01	1.6e+00	6.9e-02	1.7e-01	2.2e-01
7	0	0	0	0	0	1.1e+00	6.7e-01	1.0e-01	7.5e-02	3.8e-01
8	100	60	100	100	100	0.0e+00	8.7e-02	0.0e+00	0.0e+00	0.0e+00
9	100	88	100	100	100	0.0e+00	1.6e-01	0.0e+00	0.0e+00	0.0e+00
10	100	80	100	100	100	0.0e+00	7.2e-02	0.0e+00	0.0e+00	0.0e+00
11	20	80	4	96	100	1.3e-01	2.5e-01	9.4e-02	1.3e-01	0.0e+00
12	100	64	100	100	100	0.0e+00	6.2e+01	0.0e+00	0.0e+00	0.0e+00
13	100	76	100	100	100	0.0e+00	2.3e-01	0.0e+00	0.0e+00	0.0e+00
14	100	100	100	100	100	0.0e+00	0.0e+00	0.0e+00	0.0e+00	0.0e+00
15	100	92	100	100	100	0.0e+00	1.0e-02	0.0e+00	0.0e+00	0.0e+00
16	88	96	100	100	100	6.5e-02	6.5e-02	0.0e+00	0.0e+00	0.0e+00
17	100	100	100	100	96	0.0e+00	0.0e+00	0.0e+00	0.0e+00	1.2e-01
18	100	100	100	100	100	0.0e+00	0.0e+00	0.0e+00	0.0e+00	0.0e+00
19	100	100	100	100	100	0.0e+00	0.0e+00	0.0e+00	0.0e+00	0.0e+00
20	100	80	100	100	100	0.0e+00	5.2e-02	0.0e+00	0.0e+00	0.0e+00
21	100	100	100	100	100	0.0e+00	0.0e+00	0.0e+00	0.0e+00	0.0e+00
22	100	68	96	100	100	0.0e+00	4.0e+00	9.1e+04	0.0e+00	0.0e+00
23	100	88	100	100	100	0.0e+00	1.4e-03	0.0e+00	0.0e+00	0.0e+00
24	100	92	100	100	100	0.0e+00	9.8e-01	0.0e+00	0.0e+00	0.0e+00
25	100	100	100	88	100	0.0e+00	0.0e+00	0.0e+00	1.5e-04	0.0e+00
26	68	20	100	100	100	2.3e-02	9.3e+00	0.0e+00	0.0e+00	0.0e+00
27	100	100	100	100	100	0.0e+00	0.0e+00	0.0e+00	0.0e+00	0.0e+00
28	100	88	100	100	100	0.0e+00	2.0e-01	0.0e+00	0.0e+00	0.0e+00
29	100	100	100	100	100	0.0e+00	0.0e+00	0.0e+00	0.0e+00	0.0e+00
30	100	36	100	28	92	0.0e+00	8.1e+04	0.0e+00	2.3e+03	6.8e-02
31	100	100	100	100	100	0.0e+00	0.0e+00	0.0e+00	0.0e+00	0.0e+00
32	100	92	100	100	100	0.0e+00	1.2e-02	0.0e+00	0.0e+00	0.0e+00
33	100	100	100	100	100	0.0e+00	0.0e+00	0.0e+00	0.0e+00	0.0e+00
34	0	76	0	0	0	4.3e-01	1.3e-02	1.7e-02	4.3e-02	1.5e-01
35	0	76	0	0	0	4.9e+00	5.7e+00	1.0e-01	1.5e-01	2.0e+00
36	0	80	0	0	0	4.5e+00	2.5e-02	1.2e-01	2.9e-01	3.2e+00
37	0	40	0	0	0	2.9e-01	8.3e-02	3.6e-02	5.6e-02	2.4e-01
38	0	40	0	0	0	2.8e-01	6.3e-02	3.8e-02	5.8e-02	1.3e-01
39	0	36	0	0	0	2.8e-01	3.6e-01	3.8e-02	5.8e-02	1.3e-01
40	0	60	0	0	0	8.7e+00	4.7e-01	9.5e-01	1.9e+00	1.6e+00
41	0	100	0	0	0	9.1e+00	0.0e+00	6.7e-01	1.7e+00	7.4e+00
42	0	60	0	0	0	1.5e+01	2.9e+00	9.3e-01	2.1e+00	2.3e+00
43	0	60	0	0	0	1.4e+01	1.5e+00	1.0e+00	2.1e+00	2.8e+00
44	100	100	100	100	100	0.0e+00	0.0e+00	0.0e+00	0.0e+00	0.0e+00
45	100	92	100	100	100	0.0e+00	1.2e-02	0.0e+00	0.0e+00	0.0e+00
46	84	60	100	100	100	1.4e-02	3.0e-01	0.0e+00	0.0e+00	0.0e+00
47	92	84	100	100	100	3.8e-03	2.3e-01	0.0e+00	0.0e+00	0.0e+00
48	28	84	100	88	88	4.8e-02	2.5e-03	0.0e+00	1.3e-02	1.7e-02
49	32	60	100	100	88	2.1e-02	9.6e-04	0.0e+00	0.0e+00	1.6e-02
50	4	96	88	0	8	3.3e-02	6.7e-04	2.7e-03	7.6e-03	1.5e-02
51	0	0	0	0	0	3.8e-02	1.5e-01	2.8e-06	1.6e-05	6.8e-04
52	88	68	100	100	100	1.6e-02	1.5e-02	0.0e+00	0.0e+00	0.0e+00
53	16	4	100	100	4	3.5e-02	4.8e-02	0.0e+00	0.0e+00	5.2e-03
54	0	0	100	100	0	5.8e-02	1.0e+00	0.0e+00	0.0e+00	1.3e-03
55	0	0	24	0	0	1.2e-01	1.9e-02	2.4e-04	3.7e-03	4.9e-03
56	0	0	0	0	0	6.6e-02	2.0e-02	1.1e-03	8.3e-03	1.0e-02
57	0	4	92	0	0	1.9e-01	2.2e-02	1.5e-04	2.0e-03	1.4e-03

Table 3.2: Comparison of feasibility rate (FR) and mean value of constraint violation (MV) of the algorithms in CEC 2020 benchmark

In this figure, the logarithmic scale of the algorithms' execution time for every of the 57 problems in the 25 executions are compared. Each plot is dedicated to the problems in a unique category in this benchmark. As it is observed, EDA++ superbly overpowers the rest of the algorithms in almost all cases since it requires lower execution time. This difference is high in problems 34 to 44 and low in problems 15 to 33. It is worth to note that COLSHADE generally has the highest execution time in all of the categories except the last one, in which BP-ϵMAg-ES is the slowest algorithm.

Having the execution time and the quality of the obtained solutions, the efficiency of the algorithms can be analyzed. To this end, the following efficiency parameter is defined for each solution x:

$$\Gamma(x) = \begin{cases} 1 + \Gamma_c(x) & \Gamma_c(x) > 0 \\ \Gamma_f(x) & \Gamma_c(x) = 0 \end{cases} \tag{3.20}$$

where Γ_c and Γ_f are the scaled values of objective function and constraint violation as:

$$\Gamma_f(x) = \frac{F(x) - F_{min}}{F_{max} - F_{min}} \tag{3.21}$$

$$\Gamma_c(x) = \frac{C(x) - C_{min}}{C_{max} - C_{min}} \tag{3.22}$$

where F_{min} and F_{max} are the minimum and maximum objective values found by any of the algorithms in the competition regarding a specific problem. Similarly, C_{min} and C_{max} are the lowest and highest constraint violation, achieved by any algorithms for each problem. The defined parameters scale the objective score Γ_f and constraint score Γ_c in the interval of 0 and 1 for each solution. Having these scores, the efficiency score Γ will be a score within the interval of $0 \leq \Gamma \leq 2$. Regarding this, all feasible and infeasible solutions will be inside the interval of $0 \leq \Gamma \leq 1$ and $1 \leq \Gamma \leq 2$ respectively. Obviously $\Gamma = 0$ means that the solution is feasible with the best objective value found. If $0 < \Gamma < 1$, it shows that the solution is feasible, but it is not the best solution found in terms of the objective value. If $\Gamma = 1$, it indicates that the solution is feasible (or almost feasible) with the worst objective value in comparison to other obtained feasible solutions. If $1 < \Gamma < 2$, it shows that the solution is infeasible with constraint violation less than the worst solution found. Finally, $\Gamma = 2$ indicates that the solution is infeasible and it has the highest amount of constraint violation. The scaled execution time $\Delta(x)$ is also defined as:

$$\Delta(x) = \frac{T(x) - T_{min}}{T_{max} - T_{min}} \tag{3.23}$$

where $T(x)$ represents the execution time in obtaining the solution x, and T_{min} and T_{max} are the lowest and highest execution times between all algorithms for the corresponding problem.

The defined parameters are scaled scores, thus are independent from the problem, and can be used to analyze the algorithms considering the entire

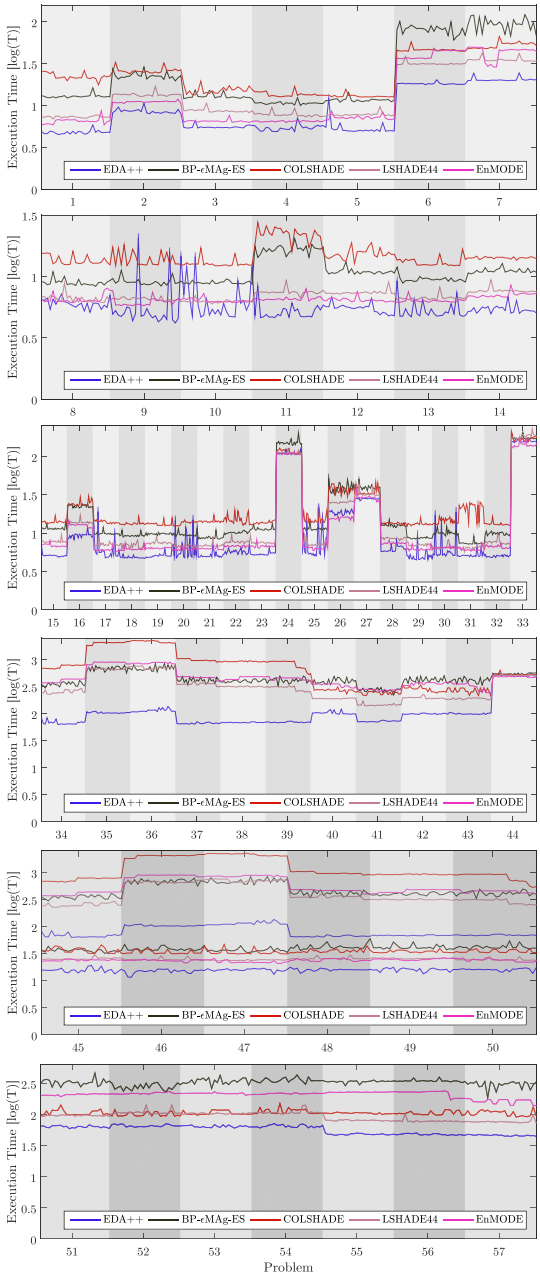

Fig. 3.8: Algorithms execution times in all repetitions

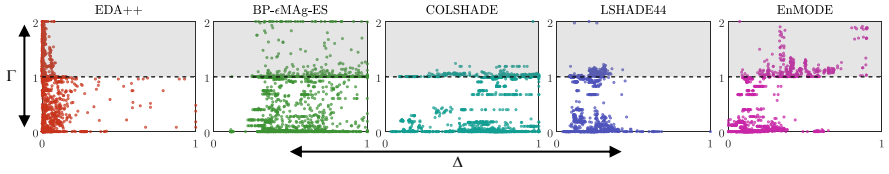

Fig. 3.9: Comparison of the algorithms' efficiency Γ vs the scaled execution time Δ

benchmark. This analysis is shown in Fig. 3.9. In this figure, each graph is dedicated to an algorithm, plotting the efficiency score and the execution time score for all of the 1425 obtained solutions (25 runs for 57 problems). Obviously, the points closer to the origin represent better solutions in both terms of quality of the solution (objective value and feasibility) and the execution time. The border between feasible and infeasible regions ($\Gamma = 1$) is marked with dashed line. Comparison of the distribution of points indicate that EDA++ has superior efficiency in comparison to the other algorithms. In this regard, EnMODE has the efficiency closest to EDA++. Also, BP-ϵMAg-ES has the highest number of points inside the feasible region. However, it generally has longer execution times in exchange for constraints' satisfaction. To be more accurate in this analysis, the Pareto set of the solutions for each problem is extracted and plotted in Fig. 3.10. The main goal of this plot is to find out the group of points that forms the Pareto sets from each algorithm. In other words, Fig. 3.10 indicates which algorithm has the highest number of dominant obtained solutions in terms of execution time and quality. As it is shown, the majority of the points correspond to EDA++. LSHADE44 has

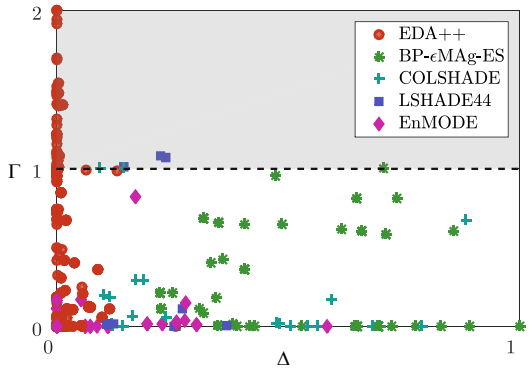

Fig. 3.10: Pareto sets of each of the 57 constrained problems from CEC2020 benchmark regarding 25 repetitions. Colored markers have been used to highlight the solutions obtained from each algorithm.

the lowest number of points within the Pareto sets, and the most competitive algorithm in comparison to EDA++ is BP-ϵMAg-ES, which has fair amount of dominant solutions with high qualities.

3.8 Simulation Results

In order to validate the effectiveness of the proposed algorithm, in this section, we provide several numerical simulations. First, a specific rendezvous mission is solved with four different initial conditions. Then, in a comparative analysis, the performance of the algorithm is compared with a Lyapunov-based approach from the literature.

The initial feasible solutions for each optimization run are obtained by solving the unconstrained version of the proposed algorithm, i.e., with feasibility conserving clustering, and mapping mechanisms disabled. In this initial run, Eq. 3.9 (the constraints) is considered as the objective function and the algorithm is run until the necessary number of initial feasible solutions are collected. Then, the proposed algorithm starts with the obtained initial seed, with all proposed mechanisms enabled. For all of the runs, the population size N_{pop} and the maximum number of generations N_{gen} are considered respectively as $20 \times n$ and $30 \times n$, where n is the number of decision variables, which itself, depends on the number of polynomial points for each case. The outlier detection distance and threshold are considered as $\lambda = 1\sigma$ and $\alpha = 0.1$ respectively.

3.8.1 Robustness Verification

For the purpose of this experiment, the rendezvous mission in an elliptical orbit with orbital elements and space mission parameters presented in as Table 3.3 is considered.

Parameter		**Value**	**Parameter**		**Value**
Semi-major axis	a	12500 km	Initial mass	m_0	170 kg
Eccentricity	e	0.4	Thrust limit	T	$\pm$120 N
Inclination	i	10°	Specific impulse	I_{sp}	300s
Right ascension	Ω	15°	Final distance	σ_r	100 m
Arg. of perigee	ω	65°	Final velocity	σ_v	0.1 m/s
True anomaly	ν	230°	Total time	t_f	5 min

Table 3.3: Mission parameters for spacecraft short-range rendezvous

As regards the disturbances, utilizing lumped periodic functions for modeling the disturbances are common in short-range rendezvous problems [185,

206]. In this scenario, the following periodic disturbance function is considered:

$$\mathbf{d} = [2\sin(0.1t)\ 2\cos(0.2t)\ 2\sin(0.3t)]' \tag{3.24}$$

In order to put the robustness of the proposed algorithm into test and also to analyze the practicality of the presented interpolation schemes, four different initial conditions are considered for this short-range rendezvous mission as in Table 3.4.

	Relative distance		**Relative velocity**	
C_1	$\mathbf{r_i} = [+1300\ +500\ -1900]\ m$		$\mathbf{v_i} = [-2.9\ -1.2\ +1.7]\ m/s$	
C_2	$\mathbf{r_i} = [+800\ -1500\ +1250]\ m$		$\mathbf{v_i} = [+7.3\ +6.8\ +1.1]\ m/s$	
C_3	$\mathbf{r_i} = [-120\ +2700\ +1430]\ m$		$\mathbf{v_i} = [+1.5\ -12.1\ -5.4]\ m/s$	
C_4	$\mathbf{r_i} = [-560\ -320\ -1790]\ m$		$\mathbf{v_i} = [-2.1\ +4.3\ -14.8]\ m/s$	

Table 3.4: Initial conditions for short-range rendezvous scenario

According to the approach presented, for each initial condition, the problem can be solved with different choices for the number of polynomial points for thrust components, mapping mechanisms, and the polynomial schemes. A number of points ranging from 5 to 24 is considered, which makes 20 choices in this regard. It will be shown that considering more points than 18 does not end in better solutions. Having three choices for interpolation schemes (SP, CR and NK) and four choices for mapping mechanisms (LD, LS, BD and BS), the total number of combinations of problem setup for each initial condition is 240. In this research, the optimization algorithm for each combination is run 10 times. In each run, the final solutions obtained are saved along with its corresponding setup. The top ten best solutions for each initial condition are sorted and presented in Table 3.5.

As can be seen, the best solutions correspond to the amount of 18 interpolation points for thrust components. The employment of other amounts of points near this number ended with, more or less, the same solutions by the optimizer. Considering a higher number of points did not help the algorithm find better solutions. Therefore, it can be concluded that there is a high possibility that the global optimal solution is reached at this point. Regarding the mapping mechanisms, it can be observed from Table 3.5 that the majority of the best solutions are found by BS mapping method regardless of the initial condition, leading to the conclusion that the most effective mapping method seems to be independent of the initial condition. Relative state vectors for the best solution for all initial conditions are illustrated in Fig. 3.11. In this figure, for each initial condition $(C_1, ..., C_4)$, the variations of relative distance and relative velocity are illustrated and the final values are displayed at the top of each plot.

C_1			C_2			C_3			C_4		
m_f	Map.	N_p	m_f	Map.	N_p	m_f	Map.	N_p	m_f	Map.	N_p
1.81272	BS	18	2.41731	BS	18	1.59422	BS	18	2.56913	BS	18
1.82008	BS	19	2.41991	BS	21	1.59428	BS	19	2.57063	LD	18
1.82061	BS	20	2.42470	BS	21	1.59774	BD	19	2.57300	BD	17
1.82092	LS	18	2.53698	BD	18	1.59804	BS	21	2.57338	LD	18
1.82363	BS	19	2.53884	BD	21	1.60652	LS	18	2.57483	BS	21
1.83340	BS	18	2.54913	LD	18	1.61041	BS	19	2.57642	LS	18
1.83557	LD	18	2.55159	BS	20	1.61129	BS	18	2.57704	BD	18
1.83562	BS	19	2.55721	BS	19	1.61427	BS	19	2.57757	LS	19
1.83741	BS	19	2.56299	LD	21	1.62225	BS	20	2.57796	BS	18
1.83764	BS	17	2.56483	LD	20	1.63085	BS	17	2.57864	LD	21

Table 3.5: Minimized fuel of top ten solutions for short-range space rendezvous with corresponding mapping mechanism and number of interpolation points

According to Fig. 3.11, the algorithm successfully achieved the best solution while satisfying the terminal conditions. The magnitude of thrust vector and the decrease in spacecraft mass within the transfer are shown in Fig. 3.12.

As shown, the chaser with initial condition C_4 needs the highest amount of fuel for space rendezvous with maximum thrust level of $122.3N$. Initial condition C_3 on the other hand, corresponds with the lowest thrust level as $45.46N$.

It is also possible to analyze the polynomial schemes within the results. The top solutions in Table 3.5 are all obtained via the SP polynomial interpolation. Best solutions obtained by other interpolation methods are presented in Table 3.6.

	C_1		C_2		C_3		C_4	
	m_f	N_p	m_f	N_p	m_f	N_p	m_f	N_p
Shape Preserving	1.81272	18	2.41731	18	1.59422	18	2.56913	18
Catmull-Rom	1.84521	20	2.64096	18	1.65746	19	2.69233	18
Not-a-Knot	1.92706	17	2.92966	20	1.93098	18	3.05119	19

Table 3.6: Best solution obtained (m_f) via different interpolation schemes

The solutions obtained indicate that SP scheme outperforms the other methods for current application in interpolating the thrust components, possibly due to less overshoot at the nodes. The thrust components corresponding to the solutions in Table 3.6 are depicted in Fig. 3.13.

In Fig. 3.13, each row of plots corresponds to one of the initial conditions, while each column is dedicated to a different interpolation scheme. It can be observed that having overshoot in splines significantly affects the solution

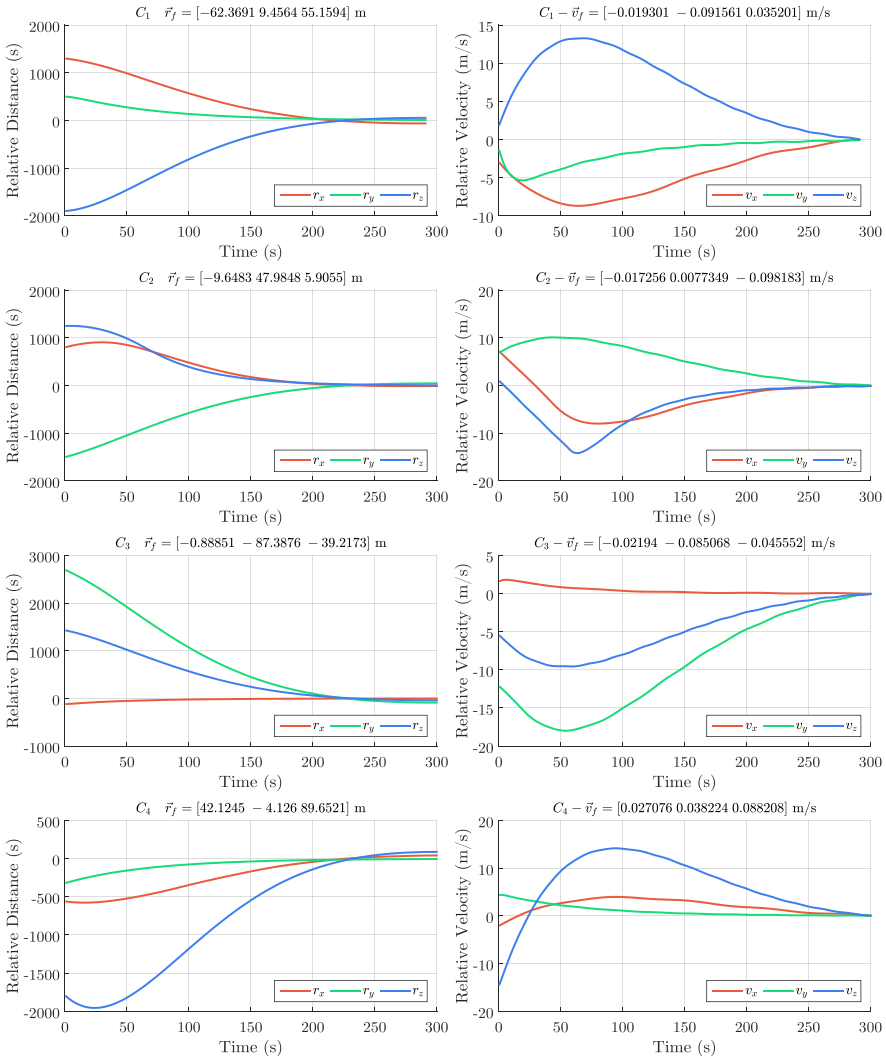

Fig. 3.11: State vectors in short-range space rendezvous for initial conditions C_1, C_2, C_3 and C_4

domain of the optimization problem, and, as a result, has a direct impact on the solution obtained by the optimizer. SP interpolation has the least overshoot, and its utilization with the presented algorithm results in feasible solutions with higher qualities.

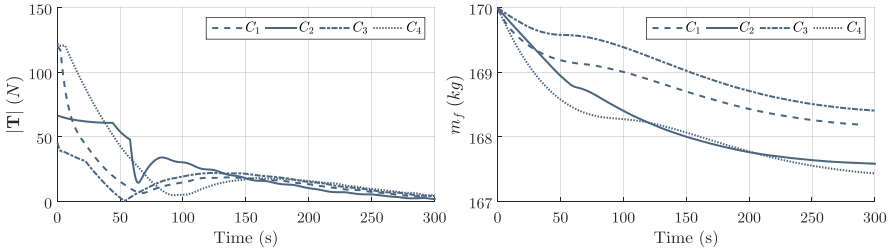

Fig. 3.12: Variation of thrust $|\mathbf{T}|$ and mass m_f in the best solutions for each initial condition

3.8.2 Comparative Results

The approach presented is compared with an analytical control approach based on implicit Lyapunov function as in [166]. The orbital parameters, the problem setup and the external disturbances are chosen accordingly. Since the specific impulse is not provided in [166], the total Δv is considered as a metric for comparison. For this problem, the optimization algorithm is utilized with respect to the four types of aforementioned mapping mechanisms, three types of polynomial schemes and ten numbers of polynomial points as $3 < N_p < 12$ (each choice of problem setup is run ten times). It is worth mentioning that by setting N_p to 1 or 2, the algorithm can not find any initial feasible solutions. On the other hand, the feasible percentage of the solution domain decreases as the number of interpolation points increases. This variation is shown in Fig. 3.14 .

As can be seen, the feasibility percentage is high when interpolation points are low. This is due to a lower number of decision variables as ($n = 3N_p$). However, the quality of the obtained solutions increases by adding more interpolation points. They are roughly illustrated in Fig. 3.15 for each optimization run. The best obtained solution based on the proposed approach corresponds to $N_p = 8$ based on SP interpolation and LS mapping scheme. Relative states for this solution are shown in Fig. 3.16.

The solution obtained is a transfer trajectory with final states as $\mathbf{r}_f = [8.6903\ 4.1487\ -1.6112]\ m$ and $\mathbf{v}_f = [0.058532\ 0.066082\ -0.046753]\ m/s$. According to the results in Fig. 3.15, only a small number of runs managed to find a solution better than the analytical solution based on the Lyapunov function. However, comparing the quality of the best solution obtained via the proposed approach as $\Delta v = 42.4854\ m/s$ with the solution associated with the Lyapunov function as $\Delta v = 44.9481\ m/s$ shows the capability of the proposed approach in finding the better solution. The variation of thrust components for two solutions is compared in Fig. 3.17.

As can be seen, the maximum thrust value is larger in the proposed approach. However, the integration of the thrust profiles, which yields the Δv, is lower than the analytical solution. Besides the comparative analysis of the

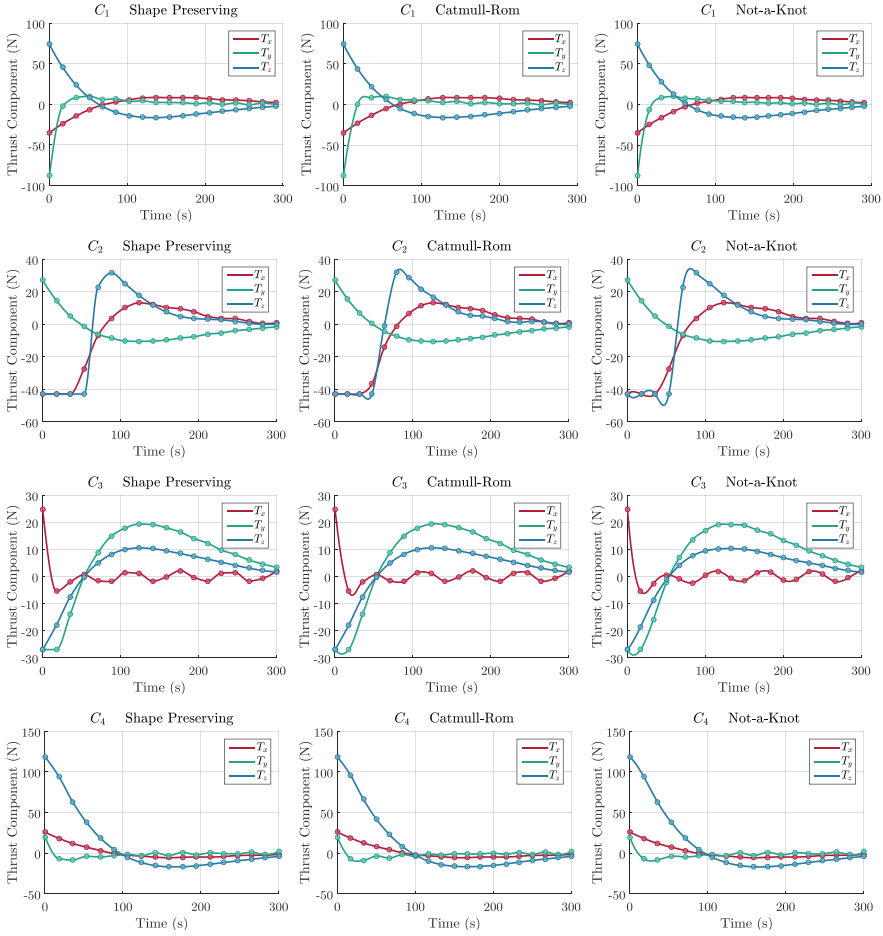

Fig. 3.13: Thrust components within the best solutions obtained for each initial condition and interpolation scheme

solutions, it is also possible to compare the performance of the proposed algorithm regarding the choices for interpolation schemes and the mapping mechanisms. This comparison is provided in Fig. 3.18.

In Fig. 3.18, each row of box plots corresponds to a unique number of interpolation points, while each column is dedicated to a specific interpolation scheme. Each box plot contains the statistical information of the quality of the solutions obtained for the proposed mapping mechanisms, along with the best objective function value found between the solutions, shown in the title. As presented, and similar to the previous simulation, SP spline has a superior advantage in finding the high quality solutions, while CR spline generally

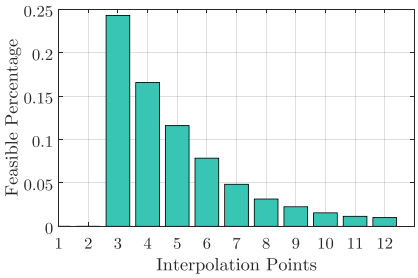

Fig. 3.14: Feasible ratio of the solution domain as the number of interpolation points

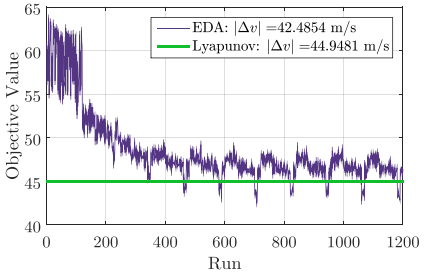

Fig. 3.15: Quality of the obtained solutions. Each point is a full execution of the algorithm

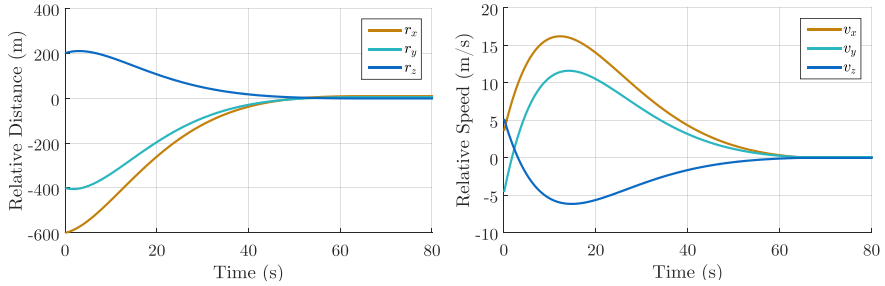

Fig. 3.16: Relative states of the chaser spacecraft $|\mathbf{r}_f| = 9.7637\ m, |\mathbf{v}_f| = 0.099\ m/s$

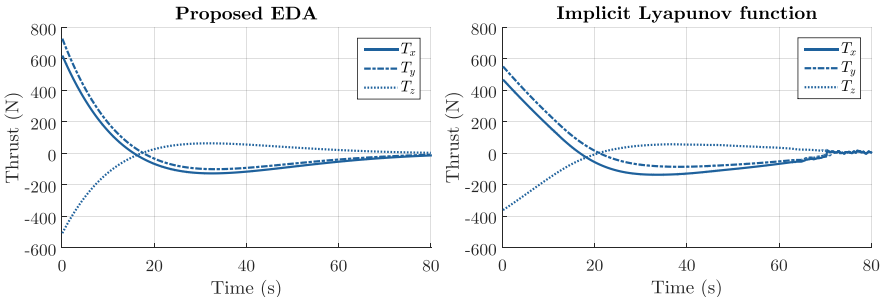

Fig. 3.17: Comparison of the thrust profiles between the proposed approach and analytical solution based on the Lyapunov function

overpowers NK splines. Also, it can be noted that increasing the number of interpolation points up to eight improves the quality of the obtained solutions. However, similar to the previous scenario, dedicating more points makes the

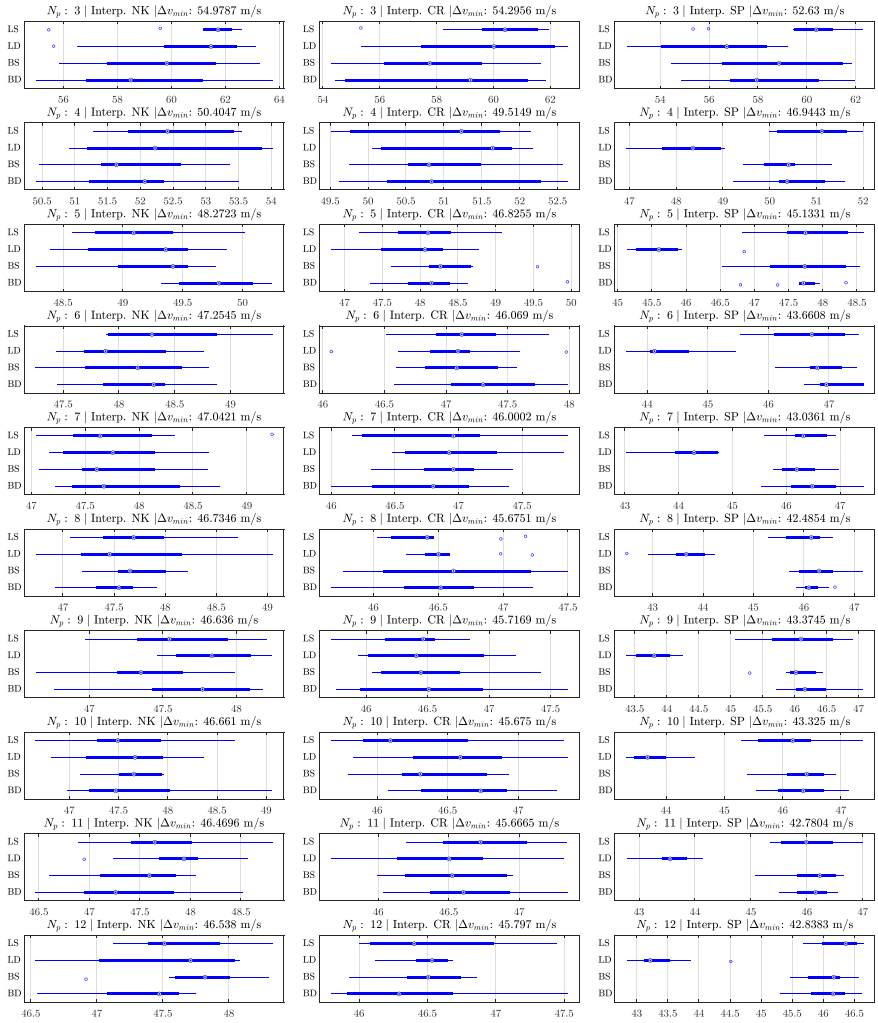

Fig. 3.18: Comparison of mapping mechanisms and interpolation schemes

algorithm find similar solutions with more effort and almost the same quality. Regarding the mapping mechanisms, there is no optimal mapping mechanism when NK and CR splines are used, due to the fact that these interpolation schemes are not the best at finding the global optimal solution. However, when SP splines are employed, LD mapping shows a significant advantage over the other mapping mechanisms.

3.9 Conclusions

In this chapter, the trajectory optimization problem for short-range space rendezvous is tackled. In this direct approach, the thrust components of the spacecraft are interpolated with various Hermite interpolation schemes. The short-range rendezvous problem turned into a black box optimization problem with nonlinear constraints. Then, an optimization algorithm based on EDAs, named EDA++ is proposed, which benefits from several feasibility conserving mechanisms.

The proposed mechanisms interact with the seeding, learning and mapping methods within the optimization process. They include a mixture with feasible centroids, outlier detection and heuristic techniques for the mapping process. Taking advantage of these mechanisms, it has been shown that the algorithm has competitive performance in comparison to other state-of-the-art algorithms in this matter. Also, it is capable of generating only feasible solutions if the initial feasible population is provided, regardless of the type of the constraints or the problem. However, there are no guarantees that other techniques reach feasible solutions. Also, the analysis shows that the proposed algorithm is generally faster than the state-of-the-art algorithms including COLSHADE, EnMODE, BP-ϵMAg-ES and LSHADE44. However, it consumes more time in comparison to traditional algorithms like GA and PSO equipped with typical constraint handling techniques. Nevertheless, the feasibility of the solutions is a fair exchange when tackling continuous optimization problems with hard constraints.

After algorithm verification, it has been applied to the short-range rendezvous trajectory optimization problem. The effort was to figure out which interpolation scheme has the advantage in finding the high quality solutions in terms of fuel mass or total Δv and how the performances of the developed mapping mechanisms are. It has been shown that SP is the most effective spline in this approach, while the best mapping method is independent of the initial condition for a specific space mission. However, when the SP splines are employed, a unique mapping mechanism exists that slightly outperforms other mapping mechanisms in finding the high quality solutions. Results also indicate that the proposed approach is robust to different initial conditions and space missions. Moreover, the presented approach managed to find solutions with higher quality in comparison to the analytical solution by implicit Lyapunov function. Improvement regarding the performance of the presented mechanisms is considered as future research.

In overall, the aim of the research was to initially fill the gap in handling constraints with some mechanisms associated with EDAs. This research can be extended to consider the vast majority of the characteristics of the proposed algorithm. One noteworthy point is that the high percentage of the time burden in the proposed method was due to the mapping process, which makes further enhancements in this mechanism in continuous domain a crucial need. Other subjects that can be considered in the future research are improving the

outlier detection methods, enhancing the seeding mechanism and analyzing various sample assignments for the components in the proposed algorithm. Future research can also be dedicated to the performance of the algorithm based on changing each of the algorithm's parameter.

4

General Conclusions and Future Work

4.1 Conclusion

Spacecraft trajectory optimization is one of the challenging subjects in astrodynamics. Cutting-edge technology for orbital maneuvers of the space vehicles turns this subject into a matter of interest for the aerospace community, as it has major impact in space mission analysis and design. At the same time, rapid development of computational capabilities allows computer science areas such as artificial intelligence to get involved with real-world problems. Within this advancement, the development of meta-heuristics and the employment of evolutionary algorithms in spacecraft trajectory optimization gained significant attention in recent years. However, no major research has been done in developing novel meta-heuristics with adaptive performance relative to the characteristics of the spacecraft trajectory optimization problems. This dissertation was an effort to make this connection and to develop novel meta-heuristics with high efficiency, concentrating on the adaptive behavior of the algorithms relative to the characteristics that describe the space orbit transfer mission. To this end, first, a comprehensive review was carried out, evaluating all state-of-art techniques and methods to deal with spacecraft trajectory optimization problems. The review was not only beneficial to choose a correct path for the research in this dissertation, but also ended up in an extensive survey, reflecting most of the research and efforts that have been carried out over the past decade while simultaneously providing a summary of the vast amount of work that was done up to this point. Then, long-range and short-range space rendezvous transfers were chosen as two general target problems to be tackled in this dissertation. Consequently, two algorithms, HSAEA and EDA++, were developed for long-range and short-range space rendezvous missions respectively. In the development of these algorithms, special attention has been paid to the discovery of the features of the problems and connecting them to the parameters of the algorithms. The algorithms were mainly developed relying on probabilistic basis. Conducted experiments confirm the robustness and the efficiency of the proposed algorithms in tackling different problems.

Besides the algorithms, a simulation software is also developed to provide a framework for the visualizing of space trajectories. The main contributions of this dissertation can be summarized as follows:

- A review for the spacecraft trajectory optimization is provided. In this review, the main process of spacecraft trajectory optimization has been decomposed into several key elements. Each element has been categorized and recent advances towards them have been identified. This review not only ended in a comprehensive survey of the literature, but also illuminated the road map for upcoming works in this matter. In this regard, it has been revealed that a new trend in this subject is designing adaptive meta-heuristics, which are useful in automatically finding and selecting good trajectory options between many possibilities one has in the preliminary phases of mission design. Also, it has been realized that it is crucial to obtain meaningful insight into whether a specific method is more suitable for solving a specific class of trajectory optimization problems than another. In other words, we found out that connecting the elements of spacecraft trajectory optimization problems and the components of algorithms is a new research line, which can lead to the development of adaptive heuristic mechanisms for dealing with complex spacecraft trajectory design problems.
- A hybrid self-adaptive evolutionary algorithm (HSAEA) was developed for solving trajectory optimization problems in long-range space rendezvous missions. The developed algorithm benefits from auto-tuning operators that work based on the complexity of the orbit transfer, and mechanisms based on Gaussian distributions for generating near-optimal transfers. The algorithm is capable of achieving orbit transfers with high quality in terms of fuel consumption and transfer time. Conducted experiments showed that the algorithm is robust to the changes in the orbital characteristics of the space mission and therefore is a practical tool in conceptual and preliminary design of space vehicles. Disabling the adaptive operators in this algorithm, makes it a hybrid optimization algorithm, capable of optimizing any types of optimization problems in continuous domain.
- An estimation of distribution algorithm based on feasibility conserving techniques (EDA++), was developed for trajectory optimization of short-range space rendezvous. Novel heuristic techniques within the framework of estimation of distribution algorithms were developed to make the algorithm compatible with non-linear constraints. The algorithm was the first attempt in developing EDAs toward constraints satisfaction in continuous optimization. It has been shown that the feasibility of the final solutions is guaranteed if the initial feasible population is available. Also, the proposed algorithm is faster than the state-of-art algorithms for optimization problems with constraints. When dealing with short-range rendezvous problems, the concentration was towards the discovery of the relation between the employed mapping mechanisms and the mathematical modeling of the

problem. Results indicate that the best mapping method is independent of the initial condition for a specific space mission. However, when specific splines are employed as the mathematical model for the thrust vector, a unique mapping mechanism exists that slightly outperforms other mapping mechanisms in finding high quality solutions. Results also indicate that the proposed algorithm is robust to different initial conditions and space missions. Moreover, the presented approach managed to find solutions with higher quality in comparison to the analytical solution by implicit Lyapunov function.

- Taking advantage of graphical user interfaces, a simulation software was developed that generates numerical data of space orbit simulation in a 3D visual representation. This software makes the trajectory design and optimization process much easier as it is incorporated with a framework that gives the scientists in astrodynamics and space engineers the ability to achieve accurate results from the simulations. This toolbox is aimed at developing a customizable application, which addresses the mission needs of satellite orbit simulations, and which analyzes scenarios considering all the perturbations influencing the orbit propagation. The results were assessed against simulations of commercially available simulation tools and were found to match satisfactorily.

4.2 Further Works

The research in this dissertation can be continued in many directions. Since the subject was a transaction between astrodynamics and evolutionary techniques, future works can be defined in either perspectives. The most important subjects that can be referred as great potentials for continuing the research are as follows:

- The review that has been carried out in this research provided a broad survey of the state-of-art methods in spacecraft trajectory optimization. In this review, it has been realized that in recent years, the development of hybrid algorithms are becoming more popular in solving spacecraft trajectory optimization problems. Similar to the survey in this dissertation, a new review of state-of-art techniques in hybridization of algorithms in spacecraft trajectory optimization is a promising area that seems to be missing in the literature. Decomposing and taxonomizing the methods and techniques in a review can lead to a comprehensive survey in this area. It is beneficial to know, which algorithms have been utilized in hybridization more and what algorithm components are usually involved in hybridization within the research in the literature.
- The other subject, which is a matter of interest, particularly in aerospace community, is comparing the features of the problems, when direct and indirect methods are combined with evolutionary algorithms. Landscape

analysis of trajectory optimization problems with direct and indirect methods, and comparing the difficulty of the problems when they are combined with evolutionary algorithms is a vast area. The analysis of the solution domain in these methods makes it possible to explore the reasons why a particular combination of a method with a specific evolutionary algorithm leads to better solution achievement.

- As for obtaining fuel-optimal and time-optimal transfers in long-range rendezvous, the research can be continued in studying the changes in the shape of the solution domain of the problems with respect to the variations of orbital elements. Further investigation in the landscape analysis of Lambert problem enables the path to develop more adaptive operators for HSAEA in dealing with long-range rendezvous trajectory optimization.
- In short-range rendezvous, similar type of research can be conducted towards the effect of various interpolation schemes in the landscape of solution domain for short-range rendezvous problems. In this dissertation, three types of Hermite splines have been adopted, and it has been shown that the employment of SP splines generally ends up in achieving solutions with higher qualities. A good potential in this area is to employ more flexible splines, such as Fourier series, which has been already shown to be quite useful in continuous thrust trajectory optimization problems [80, 81]. Extensive research in this area leads to know how much the quality of the solutions can be increased via Fourier series.
- Regarding the algorithm enhancement, the proposed EDA++ is developed with truly novel concepts within its mechanisms. Therefore, various improvements can be the aim for the future research. As for the seeding mechanism, the improvement can be towards obtaining initial feasible population in a more efficient method. For instance, the current mechanism does not use any information from the gradient of the solution domain of the constraints function. Therefore, future works can be conducted in considering gradient-based methods within the seeding mechanism to improve the process. Incorporation of such techniques, more specifically, gradient-based stochastic operators, in minimization of the objective function is also a new area for further research.
- The proposed seeding mechanism in this research performs the multivariate Gaussian EDA to minimize the constraint violation. This EDA simply includes the sequence of learning, sampling, repairing, evaluation, and replacement. However, as mentioned, it can be other EDAs as well. When having multiple constraints, one novel idea is to satisfy the constraints in a sequential manner, using the proposed EDA++. In this concept, the feasible region is discovered in different steps. In these steps, the constraints are considered one after another in the constraint violation function, and therefore the feasible region is discovered smoothly. Investigation of this concept to verify its effectiveness opens a new subject area for further research.

- In the learning mechanism, several works can be conducted for improvement. In obtaining the parent clusters within the proposed learning mechanism, the selected population is divided in a sequential manner until all of the centroids are inside the feasible region. In other words, the number of parent clusters N_c depends on the position of the centroids, which are the average value of the solutions in each cluster. However, one can consider other criteria in dividing the selected population. For instance, it is possible to check the position of the best solution instead of the average value of the solutions in each cluster in forming the parent clusters. Applying this concept makes the number of parent clusters different, which itself affects the distribution of the newly sampled solutions after the mapping process. This concept, and also similar ideas in dividing the selected population into different clusters are a vast field of research, in which lots of novel heuristic methods can be developed in the future.
- Regarding the formation of the outlier-based clusters, the two involving parameters are α and λ, representing the percentage of top solutions, and distance threshold for outlier detection respectively. These parameters are assumed fixed during optimization. However, since the main idea of creating outlier-based clusters is to compensate the covariance loss due to the mapping mechanism, it is implied that making these parameters adaptive, relative to the diversity of the solutions, will probably enhance the performance of the algorithm. Therefore, one interesting research line for developing an enhanced learning mechanism is to make these parameters adaptive according to the diversity of solutions.
- As for the mapping mechanisms, other heuristic techniques can be developed instead of the four proposed mapping methods in this research. These mechanisms perform shifting the infeasible solutions towards their respective centroid inside the feasible region. Similar to the seeding mechanism, the proposed mapping mechanisms do not use any information from the gradient of the solution domain of the constraint function. The employment of gradient-based techniques in the mapping mechanism can significantly improve the performance of the algorithm, and therefore is an important area for further works. However, more research need to be conducted in this area to realize the burden of function evaluation in such methods as well.
- Sampling new individuals from a mixture of probabilistic models has a noticeable impact on the convergence of the algorithm. In the proposed EDA++, the mixture model consists of parent clusters and outlier-based clusters, and the optimal number of sampled solutions from each components remained unknown. In this research, equal number of solutions are sampled for parent clusters and outlier-based clusters. However, in a future research, it is possible to consider adaptive number of samples for each type of clusters to create a dynamic balance between exploration and exploitation within the optimization process.

- Lastly, the simulation platform described in this dissertation can be expanded to a wider range of designs and analyses of space orbits, such as libration points and interplanetary trajectories. Given the flexibility of this application, it can be extended to perform extensive studies in orbital mechanics. Future research can also be dedicated to the development of a platform for the comparison of algorithms in terms of accuracy and computational time in optimizing a transfer trajectory.

4.3 Main Achievements

4.3.1 Journal Papers

- **Shirazi, A.**, Ceberio, J., Lozano, J. A. (2018). Spacecraft trajectory optimization: A review of models, objectives, approaches and solutions. Progress in Aerospace Sciences, 102, 76-98.
- **Shirazi, A.**, Ceberio, J., Lozano, J. A. (2019). An evolutionary discretized Lambert approach for optimal long-range rendezvous considering impulse limit. Aerospace Science and Technology, 94, 105400.
- **Shirazi, A.**, Ceberio, J., Lozano, J. A. (2020). A simulation framework for orbit propagation and space trajectory visualization. IEEE Aerospace and Electronic Systems Magazine, Accepted.
- **Shirazi, A.**, Ceberio, J., Lozano, J. A. (2020). Optimal Short-Range Space Rendezvous with Evolutionary Feasibility Conserving Techniques. Journal of Optimization Theory and Applications, Submitted.
- **Shirazi, A.**, Ceberio, J., Lozano, J. A. (2020). EDA++: Estimation of Distribution Algorithms with Feasibility Conserving Mechanisms for Constrained Continuous Optimization. IEEE Transactions on Cybernetics, Submitted.

4.3.2 Conference Papers

- **Shirazi, A.**, Ceberio, J., Lozano, J. A. (2017, June). Evolutionary algorithms to optimize low-thrust trajectory design in spacecraft orbital precession mission. In 2017 IEEE Congress on Evolutionary Computation (CEC) (pp. 1779-1786). IEEE.
- **Shirazi, A.**, Ceberio, J., Lozano, J. A. (2019, July). Optimal multi-impulse space rendezvous considering limited impulse using a discretized Lambert problem combined with evolutionary algorithms. 8th European Conference for Aeronautics and Space Sciences (EUCASS).

4.3.3 Short Visits

- 01 March-01 July 2020: Department of Aerospace Vehicles Design and Control - DCAS, ISAE-SUPAERO, France. Supervisor: Prof. Roberto Armellin.

A

Appendix

A.1 Introduction

Modeling and simulation of autonomous spacecraft have made momentous strides in recent years, and the space industry, and other aerospace professions are on the verge of being able to use computing power. The aim of this usage is to simulate reality for all kinds of applications in space engineering such as autonomy of nanosatellites [207], flight simulation [208], satellite performance simulation [209], and orbit propagation [210]. In space engineering, rapid simulation of space orbits and trajectories is essential in different aspects of space engineering including trajectory optimization [2], orbit transfers, orbit determination [211] and attitude control. Representation of the dynamical states of spacecraft while moving in an orbit is non-trivial for Earth orbiting satellites. Although preliminary analysis of satellite space orbits can be done without extensive simulation, interactive environments in simulation frameworks allow researchers to design space missions in a broad view that are difficult, expensive, or time consuming to deal with. Programming languages such as Java have been well used in different aspects of space engineering [212]. However, the amount of simulation frameworks in space engineering with MATLAB has increased in recent years. This is due to the fact that the capability of MATLAB to manage matrices is dominant and the vast number of libraries and toolboxes are available in this programming language. In recent years, several applications for orbit simulation have been developed including various MATLAB-based toolboxes. The simulation package of AGI, named Systems Tool Kit (STK) [213] is a reputed analytical framework. It has the capability of simulating dynamic environment and scheduling of events within a space mission and has a great contribution in the space engineering community. It also has the capability of integrating with MATLAB, which makes it more flexible for different types of users. The General Mission Analysis Tool (GMAT) [214] is another space mission analysis software package developed by NASA. The MATLAB interface in this software supports connections to the MATLAB environment, letting GMAT to run MATLAB scripts simultaneously.

In addition to the applications with MATLAB integration capabilities, some MATLAB toolboxes have also been developed in recent years for space orbit simulation and design. In 2015, Carrara [215] presented PROPAT, a satellite attitude and orbit analysis tool developed in MATLAB. Although the system's attitude simulation is well-developed based on attitude kinematics and dynamics, the orbit simulation process on the other hand is limited. The propagation model is based on a solution that is achieved analytically, called as Brower model [216]. Based on this model, only three orbital elements including right ascension of ascending node, argument of perigee and mean anomaly are affected by the orbital perturbations. The toolbox SPACSSIM has been introduced in [217] as another software for orbit propagation and attitude control. There are other toolboxes in this matter as well [218]. However, there is a need for a user friendly MATLAB-based application with interactive visualization capability, in which the design, simulation and trajectory optimization within the preliminary design of space systems can be tackled.

In this appendix, a simulation platform, called HOMA, is presented to simulate space trajectories around any celestial masses along with orbit analysis. The toolbox includes an orbit propagator, linked to a visualization platform with a user-friendly interface. Various orbital perturbations can be considered including Earth atmospheric drag, Earth harmonic gravity field, solar perturbations and the perturbations of other planets. Three orbit propagators are included regarding these perturbations such as two-body propagator, Simplified General Propagator (SGP4) and High Precision Orbit Propagator (HPOP). Several ordinary differential equation (ODE) solvers are considered to be used within the orbit propagation. Moreover, simulation of the satellite ground track is also included along with several scripts to perform coordinate transformation and calculation of ephemeris.

HOMA toolbox has been developed in two versions, including a MATLAB-based version and an online version [219] for space orbit simulation and analysis. Compared with other similar available MATLAB toolboxes, the MATLAB-based version benefits from a user friendly GUI and a 3D visualization platform, where the user can interact with the space orbit environment while simulating space trajectories. Rapid and instant computation and result generation make the application very practical in the preliminary design of space orbits. This appendix presents the description of the HOMA framework, as well as some simulations performed in this application. Results are validated with similar software and toolboxes.

This appendix is organized as follows. An overview of the space orbit propagation process in HOMA is provided and the main architecture of the application is described in Section A.2. Following the structure of space orbit simulation, different orbit propagation schemes in HOMA are presented in Section A.3. The environment of HOMA is introduced in Section A.4, in which the different elements of the application interface are presented. Several examples of space orbits are simulated with the application and the results are provided in Section A.5. Besides, a multi-impulse orbit transfer problem is

simulated and analyzed. This section also includes the comparison of the results with those of state-of-art toolboxes. Finally, the conclusions are provided in Section A.6.

A.2 Overall Application Scope

Simulation of space orbit trajectories involves two elements. The first is the mathematical model of the spacecraft, which describes the dynamics of the system's motion, and the second is the propagation scheme, in which the simulation process marches through the specified time. Following these bases, the overall architecture of HOMA for simulation of space orbits is illustrated in Fig. A.1.

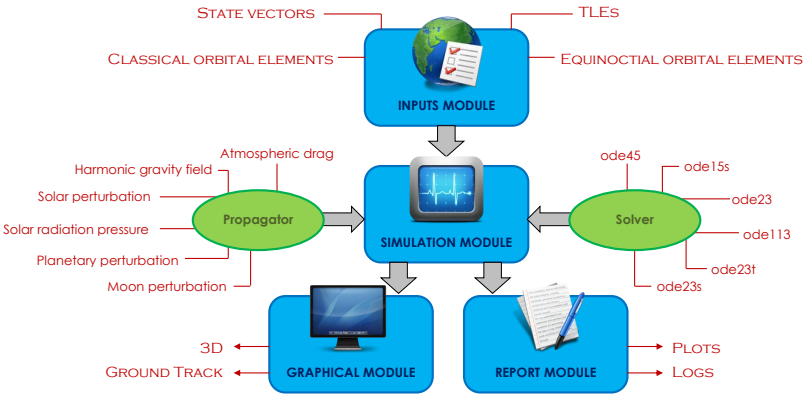

Fig. A.1: The main architecture of the simulation platform

As shown, the application has an input module in which the inputs of the simulation are defined. Generally the initial state of the spacecraft can be defined in several ways in orbital mechanics. The state vector describes the position and the velocity of the spacecraft, which can also be converted to other forms. There are two other possible representations supported in the software, including classical orbital elements and equinoctial orbital elements [26]. The difference between these representations is their practicality in space missions [220]. In addition to these representations, the well-known two-line-elements sets (TLEs) [221] can also be imported in the software in order to define the state of satellites.

The simulation module includes a propagator and a solver. In the first step of simulating the space vehicle in a space orbit, the dynamics of the spacecraft needs to be mathematically modeled. The motion of the space vehicle in orbit is defined with a set of ordinary differential equations. These equations express

a trajectory in terms of position and velocity of the space vehicle as time-dependence variables. Such mathematical model relies on having two point masses and the mutual gravitation force between these masses is taken into account [17]. In this type of mathematical model, it is assumed that the mass of the space vehicle is much smaller and negligible in comparison to central body. Having this assumption, the gravitational effect due to the mass of the space vehicle is ignored.

Considering this set of ordinary differential equations, orbit propagators are utilized to find the propagated position and velocity of the space vehicle. Various perturbations, faced by the satellite in space, are also incorporated to portray the actual behavior of the spacecraft throughout its space mission. Details regarding the orbit propagation in HOMA, such as SGP4 and HPOP propagations, are provided in the following section.

Regarding the propagation model (except for SGP4), a solver is needed to deal with the system of ODEs in several time steps. The solver part offers different ODE solvers for this where the user can specify which solver should be implemented. The choice of the solver makes it possible for the user to analyze and compare different integration methods with each other and evaluate their accuracy and computation time. It is worthy to note that the computation time varies by the propagation model when simulating several trajectories of satellites. Since the propagation process depends on the complexity of system dynamics and the stiffness of the equations, the user needs to take a balance between the expected accuracy and the dedicated time for simulation.

After the simulation process, results can be reported and illustrated in different forms including 2D plots of state variables, variation of orbital elements and satellite ground track. 3D renders of satellite motion can also be generated with different rendering options and settings as images or animations.

A.3 Spacecraft Dynamics

The process of orbit propagation for an autonomous spacecraft in HOMA is based on simulating the orbit for an extended time period using the dynamic equations of motion, models of environmental forces, torques and other physical parameters. In any space mission analysis, prediction of the orbits of satellites is an essential part and it, directly or indirectly, has impacts on the satellite's power system, attitude control, other systems [222]. The main factors affecting the orbit of a satellite are the non-spherical geometry of Earth, atmospheric drag, perturbed effects from the gravitational pull of the Sun and other planets, electromagnetic forces, radiation pressures and so forth. As stated, the software considers a number of propagators available including two-body, SGP4 and HPOP. The mathematical representation for the dynamics of the space vehicle regarding the aforementioned assumptions can be defined as the non-Keplerian two-body problem equation, [26]:

$$\ddot{\mathbf{r}} = -\frac{\mu}{r^3}\mathbf{r} + \boldsymbol{\gamma} \tag{A.1}$$

This representation is defined based on the inertial coordinate system, where $\mathbf{r}$ is the position of the space vehicle ($r = |\mathbf{r}|$), respect to the inertial coordinate frame, μ is a constant, describing the central mass gravitational property, and $\boldsymbol{\gamma}$ is the acceleration, which affects the space vehicle motion because of orbital perturbations.

A.3.1 Two-body Propagation

The simple and traditional space orbit simulation method is the two-body orbit propagation. Considering $\boldsymbol{\gamma} = 0$ in Eq. A.1, the two-body model will provide a rough idea of a spacecraft's orbit. There are two ways to simulate the spacecraft motion in this model. The first way is by solving the ODE set of equations in Eq. A.1 using iterative methods. Different ODE solvers can be utilized in this case. The second way is to simulate the satellite motion and velocity in the form of orbital elements. Since the perturbation is assumed to be zero, the motion will be on a Keplerian trajectory and the true anomaly domain will be considered instead of time for simulation. Conversion of time to true anomaly for different conic sections can be found in [15].

A.3.2 SGP4 Propagator

The SGP4 model simulates the motion of the space vehicle with a few considerations. In this model the effect of perturbations has been taken into account while computing the state vectors. The perturbations due to the shape of the Earth, the drag force due to atmosphere, Sun radiation, and the acceleration due to the gravity of other giant masses such as the Sun and Moon are involved in this model. The two-line element sets from United States Space Command are utilized for orbit propagation with SGP4 model. The general scheme of SGP4 propagator is presented in Fig. A.2 based on its available package [223]. This flowchart does not provide the details in this propagator, but it does provide an overview of the process.

The propagator includes two main steps which are the initialization and the main loop. It is coupled with many callbacks and scripts. However, the key-element functions are limited. The structure containing all the SGP4 satellite information is stored in *SatRec*. This variable will be read and updated throughout the whole process. After setting the input data, initialization will be handled first. Then a time loop updates the propagator structure and calculates the state vectors. Finally, the results will be generated according to the desired form.

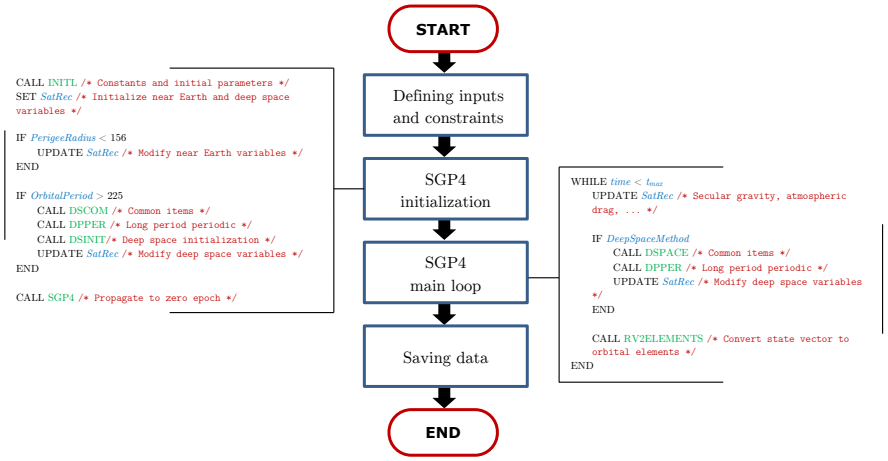

Fig. A.2: Flowchart of SGP4 propagator

A.3.2.1 Initialization

The SGP4 propagation includes several constant variables, which do not depend on time. Therefore, the propagation process starts with calculating these terms (*INITL*). This step is outlined as follows [224, 225]:

$$a_1 = (\frac{k_e}{n_0})^{\frac{2}{3}} \tag{A.2}$$

$$\delta_1 = \frac{3}{2}\frac{k_2}{{a_1}^2}\frac{(3\cos^2 i_0 - 1)}{(1-{e_0}^2)^{\frac{3}{2}}} \tag{A.3}$$

$$a_2 = a_1(1 - \frac{1}{3}\delta_1 - {\delta_1}^2 - \frac{134}{81}{\delta_1}^3) \tag{A.4}$$

$$\delta_0 = \frac{3}{2}\frac{k_2}{{a_2}^2}\frac{(3\cos^2 i_0 - 1)}{(1-{e_0}^2)^{\frac{3}{2}}} \tag{A.5}$$

$${n_0}'' = \frac{n_0}{1+\delta_0} \tag{A.6}$$

$${a_0}'' = (\frac{k_e}{{n_0}''})^{\frac{2}{3}} \tag{A.7}$$

where $k_2 = \frac{1}{2}J_2{a_E}^2$ (units of Earth radii), $J_2 = 1.082616\times10^{-3}$, $k_e = \sqrt{GM} = 0.0743669161$, G is the universal gravitational constant, M is the mass of Earth, and a_E is the equatorial radius of Earth.

Acceleration due to atmosphere drag is the next variable that is considered in initialization. The mathematical model of this term is based on a density function from power law as in [226].

$$\rho = \rho_0 \frac{(q_0 - s)^4}{(r - s)^4} \tag{A.8}$$

In this equation, r represents the distance between the space vehicle and the Earth's center, q_0 and s are the parameters for the density function. The value of q_0 is 120 km plus the radius of Earth and the value of s is the height of the spacecraft at perigee. In this model, if the perigee altitude becomes less than 156 km, the value of some of the parameters will be altered within the initialization. So the *SatRec* structure will be updated accordingly:

$$\theta = \cos(i_0) \tag{A.9}$$

$$\xi = \frac{1}{a_0 - s} \tag{A.10}$$

$$\beta_0 = (1 - {e_0}^2)^{\frac{1}{2}} \tag{A.11}$$

$$\eta = a_0 e_0 \xi \tag{A.12}$$

$$\begin{aligned} C_2 =& (q_0 - s)^4 \xi^4 n_0 (1 - \eta^2)^{-\frac{7}{2}} \times \big[a_0(1 + \frac{3}{2}\eta^2 + 4e_0\eta + e_0\eta^3) + \\ & \frac{3}{2}\frac{k_2\xi}{(1 - \eta^2)}(-\frac{1}{2} + \frac{3}{2}\theta^2)(8 + 24\eta^2 + 3\eta^4)\big] \end{aligned} \tag{A.13}$$

$$C_1 = B \times C_2 \tag{A.14}$$

$$C_3 = \frac{(q_0 - s)^4 \xi^5 A_{(3,0)} n_0 a_E \sin i_0}{k_2 e_0} \tag{A.15}$$

$$\begin{aligned} C_4 =& 2n_0(q_0 - s)^4 \xi^4 a_0 \beta_0^2 (1 - \eta^2)^{-\frac{7}{2}} \big\{[2\eta(1 + e_0\eta) + \frac{1}{2}e_0 + \frac{1}{2}\eta^3] - \\ & \frac{2k_2\xi}{a_0(1 - \eta^2)}[3(1 - 3\theta^2)(1 + \frac{3}{2}\eta^2 - 2e_0\eta - \frac{1}{2}e_0\eta^3) + \\ & \frac{3}{4}(1 - \theta^2)(2\eta^2 - e_0\eta - e_0\eta^3)\cos(2\omega_0)]\big\} \end{aligned} \tag{A.16}$$

$$C_5 = 2(q_0 - s)^4 \xi^4 a_0 \beta_0^2 (1 - \eta^2)^{-\frac{7}{2}} \big[1 + \frac{11}{4}\eta(\eta + e_0) + e_0\eta^3\big] \tag{A.17}$$

$$D_2 = 4a_0\xi {C_1}^2 \tag{A.18}$$

$$D_3 = \frac{4}{3}a_0\xi^2(17a_0 + s){C_1}^3 \tag{A.19}$$

$$D_4 = \frac{2}{3}{a_0}^2\xi^3(221a_0 + 31s){C_1}^4 \tag{A.20}$$

where $A_{3,0} = -J_3 {a_E}^3$, and $J_3 = -0.253881 \times 10^{-5}$. Zonal harmonics of the Earth are considered in initialization as:

$$\dot{M} = \Big[\frac{3k_2(-1+3\theta^2)}{2{a_0}^2{\beta_0}^3} + \frac{3{k_2}^2(13-78\theta^2+137\theta^4)}{16{a_0}^4{\beta_0}^7}\Big]n_0 \tag{A.21}$$

$$\dot{\omega} = \Big[-\frac{3k_2(1-5\theta^2)}{2{a_0}^2{\beta_0}^4} + \frac{3{k_2}^2(7-114\theta^2+395\theta^4)}{16{a_0}^4{\beta_0}^8} + \frac{5k_4(3-36\theta^2+49\theta^4)}{4{a_0}^4{\beta_0}^8}\Big]n_0 \tag{A.22}$$

$$\dot{\Omega} = \Big[\frac{3k_2\theta}{{a_0}^2{\beta_0}^4} + \frac{3{k_2}^2(4\theta^2-19\theta^3)}{2{a_0}^4{\beta_0}^8} + \frac{5k_4\theta(3-7\theta^2)}{2{a_0}^4{\beta_0}^8}\Big]n_0 \tag{A.23}$$

where $k_4 = -\frac{3}{8}J_4{a_E}^4$, and $J_4 = -1.65597 \times 10^{-6}$.

The secular coefficient and the long-period coefficient for solar and lunar gravity are also initialized as the mentioned variables are calculated. This computation includes some additional terms if the orbital period of the space vehicle is greater or equal to 225 minutes (*DSCOM*, *DPPER*, *DSINIT*). Details are provided in [227]. Lastly, when all of the initial variables are computed, the orbit is propagated once to initialize the states at epoch time.

A.3.2.2 Update and Iteration

After the initialization, the simulation goes through a loop in time domain in which the spacecraft trajectory is propagated with the constant variables calculated previously. The updating steps considers short-period and long-period effects of solar and lunar gravity, resonance effects of Earth gravity, and atmospheric drag effects.

Considering these updates, propagation yields the updated parameters in *SatRec* in each iteration. Then, the vectors representing the unit orientations are computed and the state vectors will be obtained. Knowing the state vectors, the rest of the orbital characteristics can be calculated accordingly (*RV2ELEMENTS*) as $r_k, \dot{r}_k, r\dot{f}_k, u_k, i_k$ and Ω_k in each iteration. Unit orientation vectors, as $\mathbf{U}$ and $\mathbf{V}$ are determined as:

$$\mathbf{U} = M\sin u_k + N\cos u_k \tag{A.24}$$

$$\mathbf{V} = M\cos u_k - N\sin u_k \tag{A.25}$$

where

$$\mathbf{M} = \begin{bmatrix} -\sin\Omega_k\cos i_k \\ \cos\Omega_k\cos i_k \\ \sin i_k \end{bmatrix} \tag{A.26}$$

$$\mathbf{N} = \begin{bmatrix} \cos\Omega_k \\ \sin\Omega_k \\ 0 \end{bmatrix} \tag{A.27}$$

Then, the position and velocity vectors are calculated by:

$$\mathbf{r} = r_k \mathbf{U} \tag{A.28}$$

$$\mathbf{v} = \dot{r}_k \mathbf{U} + r\dot{f}_k \mathbf{V} \tag{A.29}$$

The simplified general perturbations consider secular effects of $J2$, $J4$, and long-periodic effects of $J3$, and short-periodic effects of $J2$, along with atmospheric drag. This propagation method is very popular due to its widespread application for various kinds of missions. More detailed description of the SGP4 model as applied for the generation of NORAD 2-line elements is provided in [223].

A.3.3 High-Precision Orbit Propagator (HPOP)

The HPOP model consists of a propagation process in which the general orbital perturbations along with the gravitational forces due to other planets (N-body problem) are taken into account. As a result, this model provide more accurate prediction in comparison to other propagation models and provide a base for modeling much more complicated trajectories.

As in HPOP, the equations of acceleration in Eq. A.1 of a space vehicle are computed in the inertial reference frame as:

$$\boldsymbol{\gamma} = \boldsymbol{\gamma}_g + \boldsymbol{\gamma}_{ng} \tag{A.30}$$

where $\boldsymbol{\gamma}_g$ is the sum of the accelerations due to gravitational forces, affecting the spacecraft motion other than the typical term for Earth-gravity ($-\mu/r^3\mathbf{r}$) in Eq. A.1, and $\boldsymbol{\gamma}_{ng}$ is the sum of the non-gravitational forces, which are acted on the space vehicle surface areas.

Obviously, the main difference between this model and the two-body model is that the perturbation term is not assumed to be zero ($\boldsymbol{\gamma} \neq 0$). The main challenge in HPOP is the precise calculation of different perturbation terms. For the sake of brevity, details of these terms are omitted here and the reader is urged to refer to the references in the following sections.

The first group of terms in HPOP is the gravitational perturbations. The acceleration due to gravitational forces in this term can be expressed as follows.

$$\boldsymbol{\gamma}_g = \boldsymbol{\gamma}_{geo} + \boldsymbol{\gamma}_{st} + \boldsymbol{\gamma}_{ot} + \boldsymbol{\gamma}_{rd} + \boldsymbol{\gamma}_n + \boldsymbol{\gamma}_{rel} \tag{A.31}$$

where $\boldsymbol{\gamma}_{geo}$ is the term describing the orbital perturbations caused by the Earth's geopotential, $\boldsymbol{\gamma}_{st}$ is the perturbation caused by solid Earth tides, $\boldsymbol{\gamma}_{ot}$ is the perturbation caused by the ocean tides, $\boldsymbol{\gamma}_{rd}$ is the effect of rotational deformation, $\boldsymbol{\gamma}_n$ is the gravitational effect of other giant masses such as Sun, Moon and planets, and $\boldsymbol{\gamma}_{rel}$ is the perturbations caused by the general relativity.

The gradient of the potential function, U, that satisfies the Laplace equation, $\nabla^2 U = 0$, represents the Earth gravitational attraction. The resulting perturbation acceleration can be specified as follows:

$$\nabla U = \nabla(U_s + \Delta U_{st} + \Delta U_{ot} + \Delta U_{rd}) \tag{A.32}$$

$$= \boldsymbol{\gamma}_{geo} + \boldsymbol{\gamma}_{st} + \boldsymbol{\gamma}_{ot} + \boldsymbol{\gamma}_{rd} \tag{A.33}$$

In this equation, U_s denotes the potential caused by solid-body mass distribution. The effect of solid-body tides is represented by ΔU_{st}, ΔU_{ot} denotes the effect of potential changes due to the ocean tides, and ΔU_{rd} is the effect of the rotational deformations.

As for U_s, spherical harmonic expansion is usually utilized to express this variable, with respect to the body-fixed reference frame [228]. Also, since a non-rigid elastic body is the real formation of the Earth, its mass distribution varies with a non-uniform pattern. The solid Earth tides that affect the variation of (ΔU_{sd}) are commonly expressed by external geopotentials as described in [229]. Moreover, details regarding the computation of oceanic tidal perturbations (ΔU_{ot}) and rotational deformation (ΔU_{rd}) can be found in [230] and [231] respectively.

Acceleration due to the gravity of other planets is determined based on point mass approximations with high accuracy. The *N-body* accelerations with respect to the geocentric inertial coordinate frame, can be calculated as the following:

$$\boldsymbol{\gamma}_n = \sum_{i=1}^{N} GM_i \Big[\frac{\mathbf{r}_i}{{r_i}^3} - \frac{\boldsymbol{\Delta}_i}{{\Delta_i}^3} \Big] \tag{A.34}$$

in this equation, universal gravitational constant is represented by G, M_i denotes the mass of the *i-th* body-mass, $\mathbf{r}_i$ represents the position vector of the *i-th* body-mass and $\boldsymbol{\Delta}_i$ is the position vector of the i-th body-mass relative to the space vehicle. Details regarding the calculation of these parameters can be found in [232]. Also, the mathematical model of relativistic perturbations for Earth-orbiting spacecraft is discussed in details in [233] and [234].

Finally, the accelerations due to non-gravitational forces that affects the motion of the spacecraft are the second group of terms and can be expressed as the following.

$$\boldsymbol{\gamma}_{ng} = \boldsymbol{\gamma}_{drag} + \boldsymbol{\gamma}_{solar} + \boldsymbol{\gamma}_{Earth} + \boldsymbol{\gamma}_{thermal} \tag{A.35}$$

where $\boldsymbol{\gamma}_{drag}$ is the acceleration due to the atmospheric drag [235], $\boldsymbol{\gamma}_{solar}$ is the effect of solar radiation pressure [236], $\boldsymbol{\gamma}_{Earth}$ is the perturbations because of the radiation pressure of Earth [237], and $\boldsymbol{\gamma}_{thermal}$ is the perturbations owing to the thermal radiation [238]. All of these parameters are surface-dependent and therefore their calculations depend on the shape and orientation of the space vehicle [239].

A.4 Application Environment

Within HOMA, the user can design and construct trajectories that reflect the space mission orbit regarding the desired orbit propagation schemes as described previously. The user interface of the application is shown in Fig. A.3.

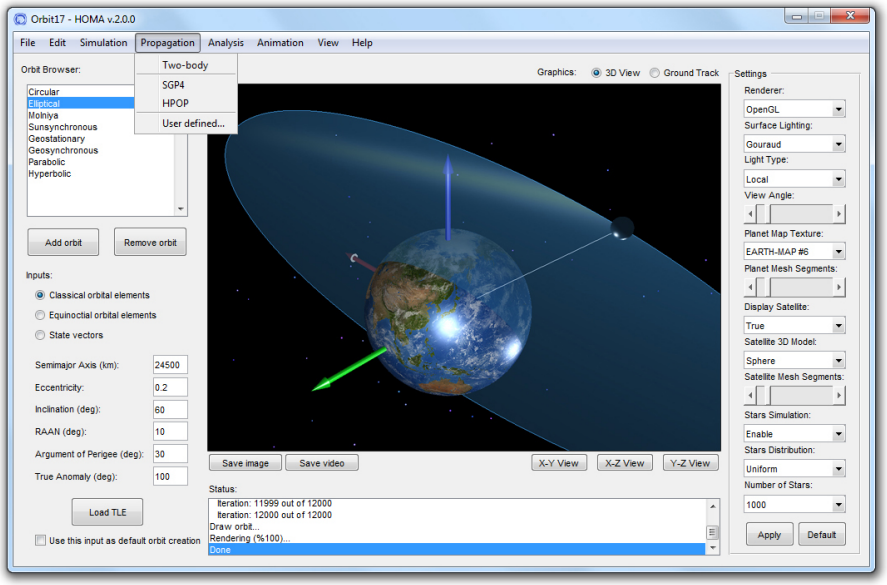

Fig. A.3: Main GUI of the HOMA toolbox

The user can specify state vectors, orbital elements or load TLEs to define the orbits. These values are then propagated in the simulation environment incorporating different perturbations according to the type of propagator. It is also possible for the user to use their own propagator within the simulation. This process may involve numerical iterations.

As Fig. A.3 shows, the orbit browser has the ability to save and keep the highly used and most frequent types of orbits which are simulated by the user. Regardless of the propagator or the solver, the application instantly converts the state vectors and orbital elements to each other. A module is also available for manually entering TLEs.

The graphic frame can be switched between two types of visualization: 3D view and ground track. In the 3D view frame, the orbit alongside the central space mass is illustrated and it enables the user to perform a number of viewing operations interactively. The camera position can be set as free or fixed with respect to a specific point relative to the spacecraft. In the ground

track frame, the ground track of the satellite is shown. Different numbers of revolutions can be specified by the user for plotting the ground track as well as various epoch times for simulation. Illustrations can be saved as high quality images or animation files while the satellite is in motion within its orbit.

The interface also has a status bar, displaying the current status of operations in the application. It is also possible to change the verbose style and save the status report accordingly. The panel for visualization settings includes several options for controlling the graphic frame. These options include the renderer type, lighting options, 3D segments control, planet textures, display options, settings for simulating stars, etc.

A.5 Simulation

In this section, several space orbit instances are simulated and the results are presented to demonstrate the performance of the application. First, the simulation capabilities of the software are demonstrated via some orbit analysis. Variation of state vectors, anomalies, and visualization of orbits have been taken into account in some space orbit examples. Second, the simulation of ground track is taken into consideration. Different geosynchronous orbits are simulated and the results are evaluated regarding the differences in orbital elements. Then, a trajectory optimization problem for long-range space rendezvous is analyzed and simulated to demonstrate the practicality of the proposed application [240]. Finally, comparisons are illustrated between the results from HOMA and two other space orbit simulation software packages including PROPAT and STK.

A.5.1 Orbit Analysis

Consider a spacecraft traveling in an orbit with semi-major axis of 14000 km and eccentricity of 0.5, inclined by 60° with argument of perigee of 10° and right ascension of ascending node of 80°. The position of the spacecraft is considered to be at a true anomaly of 170°. Visualization of this orbit based on two-body propagation with *ode45* solver is illustrated in Fig. A.4, relative to inertial frame.

As shown, the instant conversion of the orbital elements is performed within the application. This conversion renders the state vectors as $\mathbf{r} = [-3592; -20371; 0]km$ and $\mathbf{v} = [1.4471; -0.79836; -2.7085]km/s$. Some of the rendering options are also displayed in the rendered scene. As time progresses, no periodic behavior is observed with the orbital elements, but it can be observed with the state vectors since the two-body model is employed. These variations are depicted in Fig. A.5 and Fig. A.6.

In these figures, the variations of radius and velocity vectors are illustrated as functions of true anomaly. For the shown space mission, one revolution with

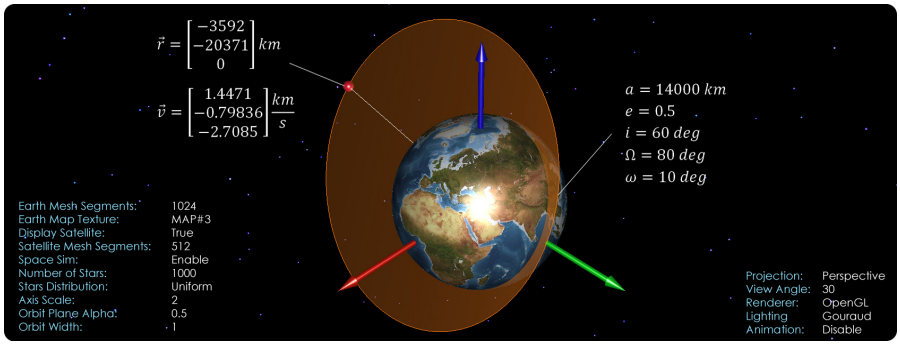

Fig. A.4: 3D visualization of a space orbit

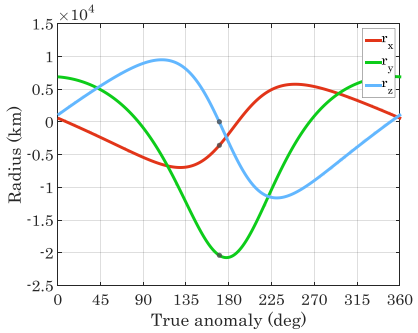

Fig. A.5: Position ($|\mathbf{r}| = 20685\ km$)

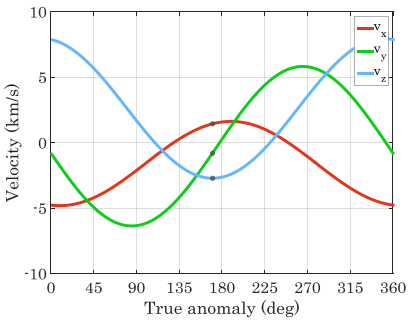

Fig. A.6: Velocity ($|\mathbf{v}| = 3.1729\ km/s$)

the orbit corresponds to the total time of 16486 seconds. While using the two-body propagation scheme, the analytical solution of the equation of relative motion is available. However, the simulation is always performed numerically in HOMA, unless specified by the user for special cases.

As another example, one shot of the ballistic view of a satellite moving in an elliptical orbit with perigee radius of 7000 km and apogee radius of 40000 km is illustrated in Fig. A.7.

As shown, it is possible to give different representations of satellite and orbit in 3D visualization frame. Within the simulation of the spacecraft motion in an elliptical space orbit, one can not picture mean anomaly as a physical element of the motion as the eccentric anomaly; rather it is related to time. However, eccentric and mean anomalies can be calculated as functions of each other. Mean, eccentric, and true anomaly are all equal to zero at perigee. Likewise, all three elements are equal to 180° at apogee. These variations are depicted in Fig. A.8 for the an orbit with a given eccentricity.

Fig. A.8 also represents the flight path angle. This angle is simply the one that the local horizon makes with the velocity vector $\mathbf{v}$ which is normal

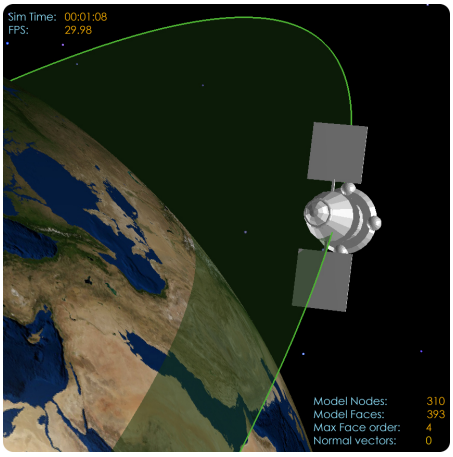

Fig. A.7: Orbit visualization (ballistic view)

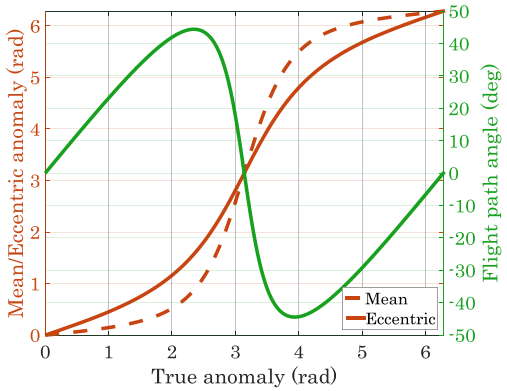

Fig. A.8: Variation of flight path angle, mean and eccentric anomalies ($e = 0.7$)

to the position vector. The sign of this angle is positive as the space vehicle travels away from perigee and is negative as the space vehicle is approaching the perigee point. This angle plays an important role while calculating the required velocity increment within an orbit transfer. When a Δv is computed for an impulsive orbit transfer at a point that is not on the apse line, including the variations of direction is an important matter as well as the magnitude of the velocity vector. The difference in the magnitude of the two vectors identified the change in the velocity, and the difference in the flight path angles shows the change in the direction.

A.5.2 Ground Track

Simulation of the ground track is one of the key elements in orbital mechanics, specifically for space mission design and tracking control [26]. It helps to generate appropriate track distance and revisit frequency over a given area. Different approaches in space engineering are based on ground track analysis, which permits the construction of ground track patterns and the determination of satellite arrangements. One sample representation is illustrated in Fig. A.9 for a space orbit with semimajor axis of 28000 km and eccentricity of 0.3, inclined by 60°.

In Fig. A.9, the projection of a satellite's orbit onto the Earth's surface is plotted for 5 days of satellite motion. As can be seen, the satellite reaches a maximum and minimum amplitude during each revolution while passing over the equator twice, therefore on a mercator projection, the ground track of the satellite resembles a sine-like curve. Since the Earth rotates eastward beneath the satellite orbit at 15.04 deg/h, the ground track advances westward at that rate.

Another family of orbits which is a matter of interest while examining the satellite ground tracks is the geosynchronous orbits. The elliptical geosynchronous orbits create drifts east and west as the spacecraft travels faster or slower at different points on its trajectory. Different combinations of inclination and eccentricity yields a motion relative to a fixed point on the ground track. Fig. A.10 illustrates three instances of geosynchronous orbit ground tracks.

The eight-like ground track (green line) is that of the geosynchronous orbit with no eccentricity. Obviously, changing the inclination of this orbit to zero

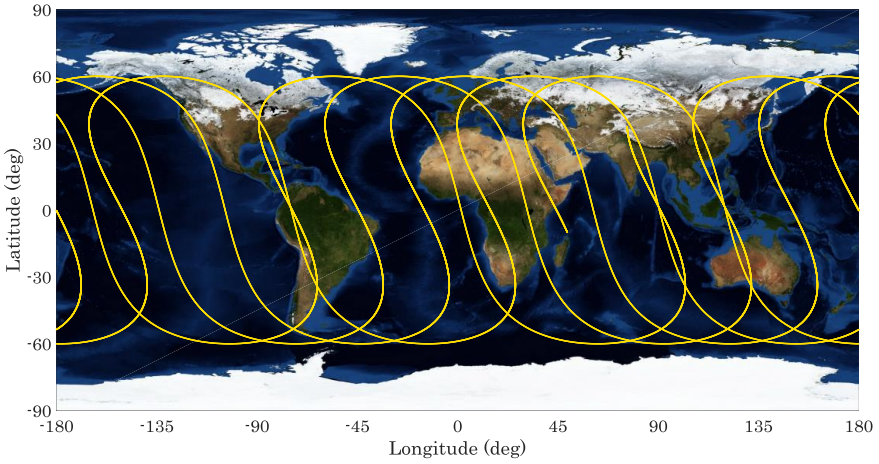

Fig. A.9: Satellite ground track ($a = 28000\ km, e = 0.3, i = 60°, \Omega = 30°, \omega = 190°$)

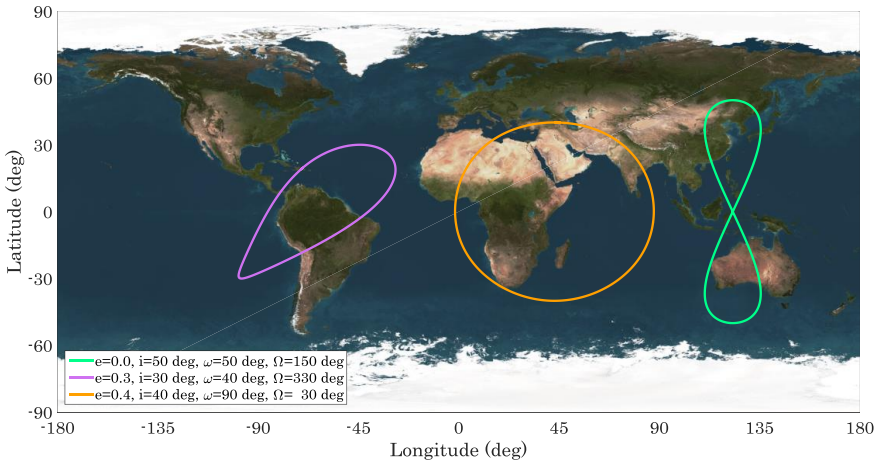

Fig. A.10: Geosynchronous ground tracks

makes this geosynchronous a geostationary orbit in which the satellite sits fixed at the crossover point of the eight-like shape (over the equator). On the other hand, if the eccentricity is increased, it results in a slanted teardrop shape. Typically, eccentric geosynchronous orbits will result in a slanted figure-eight. It just happens to have the crossover point at the northern apex of the ground track. Combining these modifications with various argument of perigees results different coverage areas for the geosynchronous orbits (orange and purple lines).

A.5.3 Autonomous Space Rendezvous

The presented framework is capable of linking with various optimization algorithms for solving spacecraft trajectory optimization problems such as interplanetary transfers [241] and space rendezvous [242]. In this section, a multi-impulse trajectory optimization problem is optimized and simulated to illustrate the capability of this tool. An evolutionary algorithm based on a discretized Lambert problem as described in [240] is linked to the platform to solve a long-range space rendezvous problem considering impulse limit.

In a rendezvous maneuver, two orbiting vehicles observe one another from each of their own free-falling, rotating, clearly non-inertial frames of reference. A rendezvous maneuver usually involves a target space vehicle, which is passive and non-maneuvering, and a chase spacecraft, which is active and performs the maneuvers required to bring itself alongside the target. In the long-range space rendezvous, the chaser executes several maneuvers under the guidance of the ground telemetry tracking and command network, so that the navigation sensors of the chaser can catch the target. The major objectives of this phase include adjusting the phase angle between the two spacecraft,

reducing the orbital plane differences, increasing the orbital height, and initiating the relative navigation. The initial states of the chaser and the target spacecraft for the selected mission are provided in Table A.1.

Table A.1: Initial states of two spacecraft in long-range space rendezvous

Orbital elements	Chaser	Target
a *(km)*	7000	30000
e	0.0	0.1
i *(deg)*	60	0
Ω *(deg)*	0	80
ω *(deg)*	205	225
θ *(deg)*	3	120

The optimal sequence of impulses is obtained and transfer trajectories are simulated within the proposed framework. 3D illustration of the transfer trajectories are depicted in Fig. A.11.

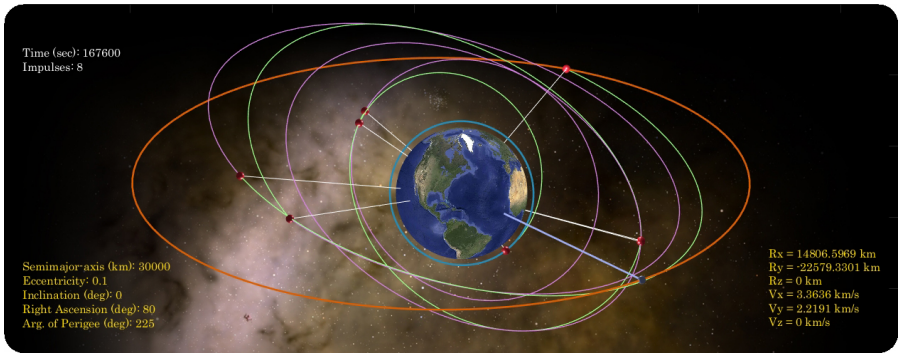

Fig. A.11: 3D visualization of multi-impulse long-range space rendezvous

In this figure, the initial and final orbits are illustrated along with coasting trajectories between the impulses during the space mission. In this scenario, eight impulses fulfill the orbital maneuver with respect to the mission criteria. The location and the corresponding radius of the impulses are shown in the figure, indicating the anomalies of the space vehicle at the intersection of transfer trajectories. Details regarding the variation of orbital elements due to the act of impulses are shown in Fig. A.12.

As it is shown, the changes of five orbital elements, including semi-major axis (a), eccentricity (e), inclination (i), right ascension of ascending node (Ω), argument of perigee (ω), along with true anomaly indicates the time of the impulses acted on the chaser spacecraft. The fluctuations within the

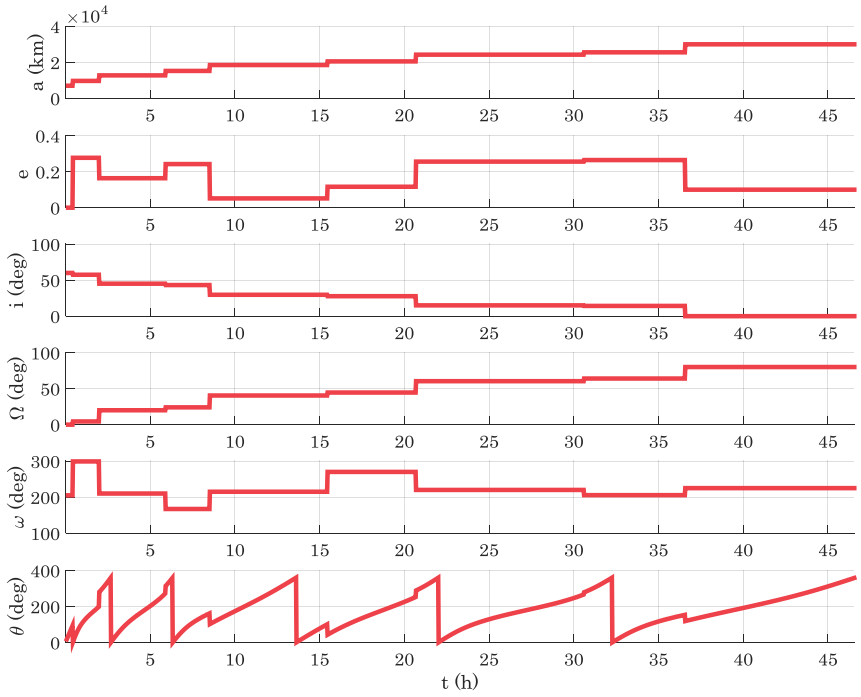

Fig. A.12: Variation of orbital elements of the chaser spacecraft

plots are due to the fact that when the impulse is acted on the chaser, the orbital plane and the shape of the orbit is changed, placing the chaser on a new trajectory during the process of reaching the target space vehicle. These variations correspond with the changes of the state vectors of the chaser, including the position ($\mathbf{r}$) and the velocity ($\mathbf{v}$) of the vehicle as shown in Fig. A.13 and Fig. A.14. Similarly, non-continuity in the velocity components are due to the fact that the maneuvers are impulsive, not with continuous thrust.

As the figures indicate, the state vectors of the chaser converge to the states of the target as the spacecraft travels on the coasting trajectories. According to the simulation results, the time between two sequential impulses, which increases as the chaser approaches the target is increasing. It agrees with the fact that since the orbital period of the target orbit is more than the initial orbit, the chaser transfers to orbits with higher angular velocities. Fig. A.15 indicates the location of the impulses on the map.

In this figure, the ground track of the chaser within the orbital maneuver is also illustrated. The size of the red circles corresponds to the magnitude of impulses in each location. Each impulse is acted on the chaser in a different altitude, as detailed in Table A.2.

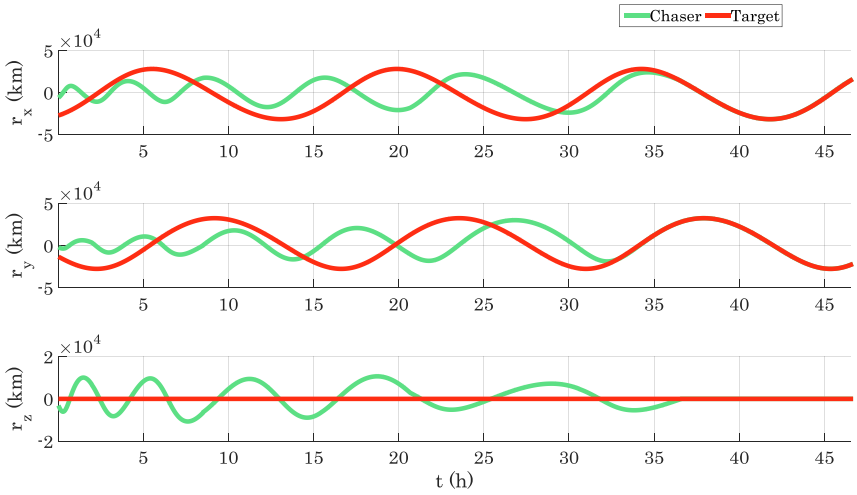

Fig. A.13: Position states of the chaser and the target spacecraft

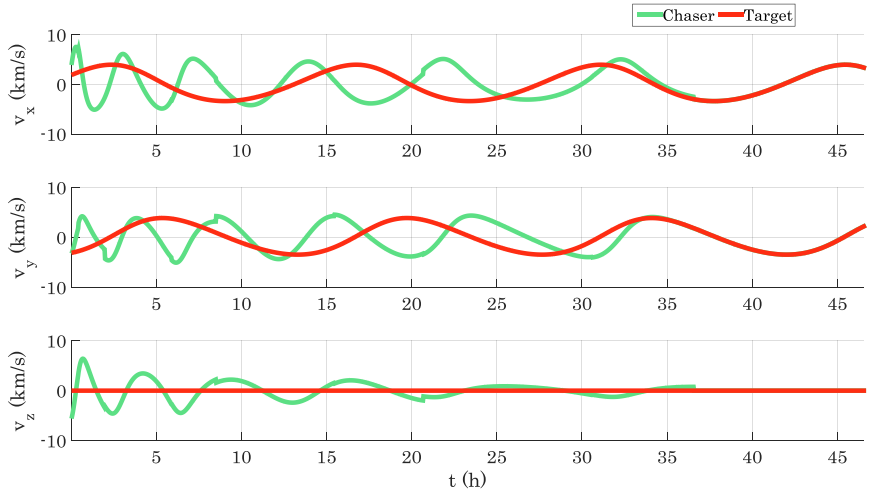

Fig. A.14: Velocity states of the chaser and the target spacecraft

According to the results, the chaser starts its first transfer in its initial trajectory, which is an inclined orbit. As the maneuver proceeds, the impulses occur at higher altitudes, which makes the distance between target and chaser decrease. After the final impulse, the chaser is at the same position and velocity of the target where the orbit inclination becomes zero.

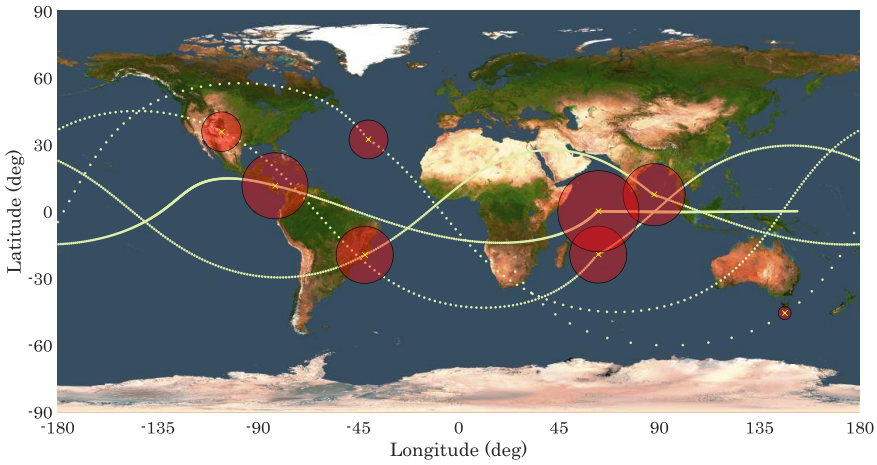

Fig. A.15: Ground track of the chaser and the location of impulses

Table A.2: Geographic coordinates of the impulses acted on the chaser

Impulse	Time (s)	Latitude (deg)	Longitude (deg)	Altitude (km)
1	1600	-45.5765	146.2579	632.7572689
2	7200	32.33925	-40.4453	5670.861950
3	21300	35.80216	-106.418	5899.997229
4	30700	-19.3534	62.62282	12282.44339
5	55700	-19.5134	-42.1559	12285.10654
6	74500	7.51914	87.68698	14559.53319
7	110200	11.53168	-82.4076	16475.49403
8	131700	0.105248	62.70357	24837.11411

A.5.4 Results Validation

Several verification tests are performed for validating the simulation results in HOMA. The orbit propagation process is the main subject which has been verified through different comparisons. The two perturbed orbit propagation models, SGP4 and HPOP are considered for validation. Results are compared with the outputs of two similar simulation platforms, STK by Analytical Graphics, Inc. and PROPAT by Carrara. While the SGP4 propagation

Table A.3: IRIDIUM 162 two-line element (June 26th 2018)

1	43482U 18047G 18177.64341180 +.00000099 +00000-0 +23569-4 0 9999
2	43482 086.4507 225.5608 0002381 102.6517 257.4950 14.43275512005031

scheme is tested with the former toolbox, the HPOP propagation model is compared with the latter framework.

The two-line element set in Table A.3 is considered as the input for SGP4 propagation in STK as well as in HOMA. The TLE is given for IRIDIUM 162, retrieved from CelesTrak [243]. The satellite is a part of the Iridium constellation, launched in May 2018. The simulation results are illustrated in Fig. A.16 to A.21. Each figure shows the variation of one orbital elements and the variations are depicted for HOMA and STK respectively. Also, the maximum value of absolute error, (E), for each quantity is extracted from the figures.

Since the SGP4 model is used here, the orbital elements are not constant as the time passes. According to the comparison, the overall shape of the outputs are the same, leading to conclude that the perturbed trajectory is formulated properly regarding the SGP4 orbit propagation model. The differences are also negligible regarding the type of each element, and acceptable for preliminary space orbit simulation and design as in typical space missions.

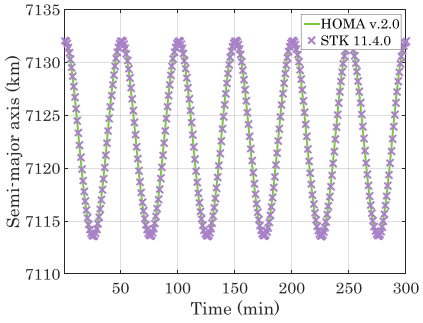

Fig. A.16: Semi-major axis ($E_a = 6.4m$)

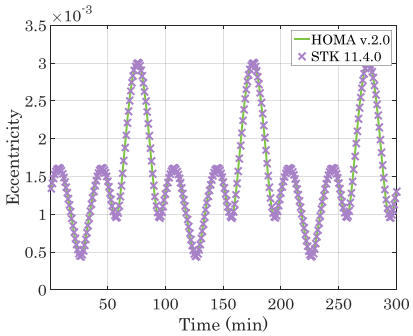

Fig. A.17: Eccentricity ($E_e = 1.38e{-}6$)

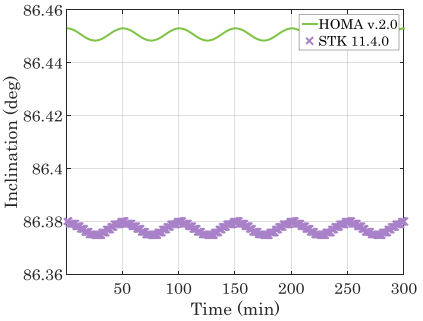

Fig. A.18: Inclination ($E_i = 0.073°$)

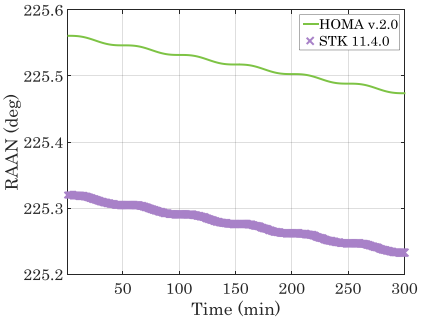

Fig. A.19: RAAN ($E_\Omega = 0.241°$)

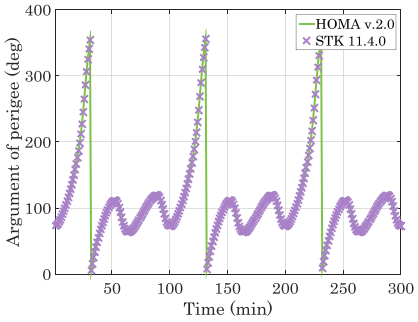

Fig. A.20: Arg. of perigee ($E_\omega = 0.142°$)

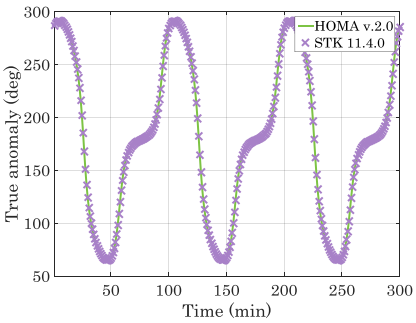

Fig. A.21: True anomaly ($E_\theta = 0.072°$)

The results regarding the high precision orbit propagation are compared with the Brower model in PROPAT, in which the gravitational field of the Earth along with the flattening of the poles are considered. Considering an elliptical orbit with semi-major axis of 19000 km and eccentricity of 0.45, inclined by 65° with right ascension of ascending node of 5° and argument of perigee of 25°. The state vectors of the satellite at perigee is $\mathbf{r} = [9489.07 \quad 2507.12 \quad 3598.32]km$ and $\mathbf{v} = [-3.16 \quad 2.62 \quad 6.19]km/s$ in the ECI frame. Propagation of space orbit is done within 10 uniform time steps. The relative percentage errors ($\hat{E}$) for state variables with respect to PROPAT results are tabulated in Table A.4.

As can be seen, the maximum value of relative percentage error is within the order of 10^{-2} after one period. Note that the orbital period of the selected orbit is 26064 seconds.

Table A.4: Relative percentage error of state vectors

Time (s)	$\hat{E}_{r_x}$	$\hat{E}_{r_y}$	$\hat{E}_{r_z}$	$\hat{E}_{v_x}$	$\hat{E}_{v_y}$	$\hat{E}_{v_z}$
2600	2.19E-04	3.56E-04	1.13E-04	1.92E-04	1.24E-02	3.96E-04
5200	8.55E-05	1.22E-03	2.56E-04	1.76E-04	5.65E-04	3.39E-04
7800	1.54E-04	1.32E-02	1.35E-03	9.07E-04	1.54E-05	3.27E-04
10400	3.60E-04	1.54E-04	2.45E-03	1.51E-02	4.35E-05	1.82E-04
13000	7.57E-04	2.00E-04	8.74E-04	3.46E-03	3.71E-04	1.95E-04
15600	1.55E-03	1.34E-04	3.87E-04	2.14E-03	2.22E-03	1.27E-03
18200	3.56E-03	3.05E-04	1.70E-04	1.61E-03	7.14E-02	7.60E-03
20800	1.75E-02	1.57E-03	1.31E-03	8.14E-04	7.68E-03	1.20E-02
23400	8.17E-03	6.80E-03	5.46E-03	3.36E-03	4.82E-03	5.94E-03
26000	3.96E-03	2.74E-02	5.78E-02	2.66E-02	3.83E-03	1.79E-03

References

1. K. Hussain, M. N. M. Salleh, S. Cheng, and Y. Shi, "Metaheuristic research: a comprehensive survey," *Artificial Intelligence Review*, vol. 52, no. 4, pp. 2191–2233, 2019.
2. A. Shirazi, J. Ceberio, and J. A. Lozano, "Spacecraft trajectory optimization: A review of models, objectives, approaches and solutions," *Progress in Aerospace Sciences*, vol. 102, pp. 76–98, oct 2018.
3. J. T. Betts, "Survey of numerical methods for trajectory optimization," *Journal of Guidance, Control, and Dynamics*, vol. 21, pp. 193–207, mar 1998.
4. B. A. Conway, "A survey of methods available for the numerical optimization of continuous dynamic systems," *Journal of Optimization Theory and Applications*, vol. 152, pp. 271–306, Feb 2012.
5. P. A. de Sousa-Silva and M. O. Terra, "A survey of different classes of Earth-to-Moon trajectories in the patched three-body approach," *Acta Astronautica*, vol. 123, pp. 340 – 349, 2016. Special Section: Selected Papers from the International Workshop on Satellite Constellations and Formation Flying 2015.
6. Y. Luo, J. Zhang, and G. Tang, "Survey of orbital dynamics and control of space rendezvous," *Chinese Journal of Aeronautics*, vol. 27, pp. 1–11, feb 2014.
7. S. N. D'Souza and N. Sarigul-Klijn, "Survey of planetary entry guidance algorithms," *Progress in Aerospace Sciences*, vol. 68, pp. 64–74, jul 2014.
8. M. Shirobokov, S. Trofimov, and M. Ovchinnikov, "Survey of station-keeping techniques for libration point orbits," *Journal of Guidance, Control, and Dynamics*, vol. 40, pp. 1085–1105, may 2017.
9. M. Xu, Y. Liang, and K. Ren, "Survey on advances in orbital dynamics and control for libration point orbits," *Progress in Aerospace Sciences*, vol. 82, pp. 24–35, apr 2016.
10. G. Huang, Y. Lu, and Y. Nan, "A survey of numerical algorithms for trajectory optimization of flight vehicles," *Science China Technological Sciences*, vol. 55, pp. 2538–2560, Sep 2012.
11. J. T. Betts, *Practical methods for optimal control and estimation using nonlinear programming*. SIAM, 2010.
12. B. A. Conway, *Spacecraft trajectory optimization*, vol. 29. Cambridge University Press, 2010.
13. I. M. Ross, *A primer on Pontryagin's principle in optimal control*. Collegiate publishers, 2 ed., 2015.

14. D. E. Kirk, *Optimal control theory: an introduction*. Courier Corporation, 2012.
15. H. D. Curtis, *Orbital mechanics for engineering students*. Butterworth-Heinemann, 2014.
16. J. Sims and S. Flanagan, "Preliminary design of low-thrust interplanetary missions," *No. AAS 99-338, AAS/AIAA Astrodynamics Specialist Conference*, 1999.
17. W. E. Wiesel, ed., *Space Flight Dynamics*. CreateSpace Independent Publishing Platform, 2010.
18. X. Y. Jiang, Y. J. Lian, H. B. Zhang, and G. J. Tang, "A multi-impulse extended method for low-thrust trajectory optimization," in *2013 6th International Conference on Recent Advances in Space Technologies (RAST)*, pp. 365–368, June 2013.
19. G. Lantoine, *Methodology for robust optimization of low-thrust trajectories in multi-body environments*. PhD thesis, Georgia Institute of Technology, 2010.
20. K. Sankaran, E. Y. Choueiri, and S. C. Jardin, "Comparison of simulated magnetoplasmadynamic thruster flowfields to experimental measurements," *Journal of Propulsion and Power*, vol. 21, no. 1, pp. 129–138, 2005.
21. J. Dankanich, "Low-thrust mission design and application," in *46th AIAA/ASME/SAE/ASEE Joint Propulsion Conference & Exhibit*, pp. 1–19, American Institute of Aeronautics and Astronautics, jul 2010.
22. I. D. Boyd, "Numerical modeling of spacecraft electric propulsion thrusters," *Progress in Aerospace Sciences*, vol. 41, pp. 669–687, nov 2005.
23. E. Taheri, *Rapid space trajectory generation using a Fourier series shape-based approach*. Michigan Technological University, 2014.
24. V. V. Ivashkin and I. V. Krylov, "Optimization of transfer trajectories to the apophis asteroid for spacecraft with high and low thrust," *Cosmic Research*, vol. 52, pp. 106–117, mar 2014.
25. P. Gurfil, *Modern astrodynamics*, vol. 1. Butterworth-Heinemann, 2006.
26. D. A. Vallado, *Fundamentals of Astrodynamics and Applications, 4th Ed.* Microcosm Press, 2013.
27. S. Li, Y. Zhu, and Y. Wang, "Rapid design and optimization of low-thrust rendezvous/interception trajectory for asteroid deflection missions," *Advances in Space Research*, vol. 53, pp. 696–707, feb 2014.
28. A. Shirazi, J. Ceberio, and J. A. Lozano, "Evolutionary algorithms to optimize low-thrust trajectory design in spacecraft orbital precession mission," in *2017 IEEE Congress on Evolutionary Computation (CEC)*, pp. 1779–1786, IEEE, jun 2017.
29. J. L. Junkins, A. B. Younes, R. M. Woollands, and X. Bai, "Picard iteration, chebyshev polynomials and chebyshev-picard methods: Application in astrodynamics," *The Journal of the Astronautical Sciences*, vol. 60, pp. 623–653, dec 2013.
30. D. J. Gondelach and R. Noomen, "Hodographic-shaping method for low-thrust interplanetary trajectory design," *Journal of Spacecraft and Rockets*, vol. 52, no. 3, pp. 728–738, 2015.
31. C. Xie, G. Zhang, and Y. Zhang, "Shaping approximation for low-thrust trajectories with large out-of-plane motion," *Journal of Guidance, Control, and Dynamics*, vol. 39, pp. 2780–2789, dec 2016.
32. W. E. Wiesel, *Modern Astrodynamics*. CreateSpace Independent Publishing Platform, 2010.

33. H. Shang, H. Cui, P. Cui, and E. Luan, "Optimal-fuel, low-thrust Earth-Ivar transfer trajectory with Venus gravity assist," in *2006 1st International Symposium on Systems and Control in Aerospace and Astronautics*, pp. 1–5, IEEE, 2006.
34. H. Shang, P. Cui, E. Luan, and D. Qiao, "Study of design and optimization of low-thrust transfer trajectory with planetary aerogravity assist," in *Computational Engineering in Systems Applications, IMACS Multiconference on*, vol. 1, pp. 690–695, IEEE, 2006.
35. A. Peloni, C. R. McInnes, and M. Ceriotti, "Osculating keplerian elements for highly non-keplerian orbits," in *27th AAS/AIAA Space Flight Mechanics Meeting*, pp. 1–20, 2017.
36. G. R. Hintz, "Survey of orbit element sets," *Journal of Guidance, Control, and Dynamics*, vol. 31, pp. 785–790, may 2008.
37. G. R. Hintz, *Orbital Mechanics and Astrodynamics*. Springer International Publishing, 2015.
38. J. Li, S. R. Vadali, and H. Baoyin, "Autonomous lunar orbit rendezvous guidance based on j2-perturbed state transition matrix," *Journal of Guidance, Control, and Dynamics*, vol. 38, no. 2, pp. 315–322, 2015.
39. Y.-Z. Luo, G.-J. Tang, and H. yang Li, "Optimization of multiple-impulse minimum-time rendezvous with impulse constraints using a hybrid genetic algorithm," *Aerospace Science and Technology*, vol. 10, pp. 534–540, sep 2006.
40. R. Whitley and R. Martinez, "Options for staging orbits in cislunar space," in *Aerospace Conference, 2016 IEEE*, pp. 1–9, IEEE, 2016.
41. D. Guzzetti, E. M. Zimovan, K. C. Howell, and D. C. Davis, "Stationkeeping analysis for spacecraft in lunar near rectilinear halo orbits," in *27th AAS/AIAA Space Flight Mechanics Meeting*, pp. 1–20, 2017.
42. A. J. Abraham, D. B. Spencer, and T. J. Hart, "Early mission design of transfers to halo orbits via particle swarm optimization," *The Journal of the Astronautical Sciences*, vol. 63, pp. 81–102, Jun 2016.
43. Y. Ulybyshev, "Long-term station keeping of space station in lunar halo orbits," *Journal of Guidance, Control, and Dynamics*, vol. 38, pp. 1063–1070, jun 2015.
44. G. Gomez, M. W. Lo, and J. J. Masdemont, *Libration Point Orbits and Applications*. World Scientific, 2002.
45. R. Marler and J. Arora, "Survey of multi-objective optimization methods for engineering," *Structural and Multidisciplinary Optimization*, vol. 26, pp. 369–395, apr 2004.
46. B. Derbel, D. Brockhoff, A. Liefooghe, and S. Verel, "On the impact of multiobjective scalarizing functions," in *International Conference on Parallel Problem Solving from Nature*, pp. 548–558, Springer, 2014.
47. M. K. Fain and O. L. Starinova, "Ballistic optimization of the L1-L2 and L2-L1 low thrust transfers in the Earth-Moon system," in *2015 7th International Conference on Recent Advances in Space Technologies (RAST)*, pp. 95–98, June 2015.
48. Y. Wang, M. Zhu, Y. Wei, and Y. Zhang, "Solar sail spacecraft trajectory optimization based on improved imperialist competitive algorithm," in *Intelligent Control and Automation (WCICA), 2012 10th World Congress on*, pp. 191–195, IEEE, 2012.
49. M. Vasile and M. Locatelli, "A hybrid multiagent approach for global trajectory optimization," *Journal of Global Optimization*, vol. 44, pp. 461–479, Aug 2009.

50. O. Abdelkhalik and D. Mortari, "N-impulse orbit transfer using genetic algorithms," *Journal of Spacecraft and Rockets*, vol. 44, pp. 456–460, mar 2007.
51. J. Zhao and R. Zhou, "Pigeon-inspired optimization applied to constrained gliding trajectories," *Nonlinear Dynamics*, vol. 82, pp. 1781–1795, jul 2015.
52. N. Yokoyama and S. Suzuki, "Modified genetic algorithm for constrained trajectory optimization," *Journal of Guidance, Control, and Dynamics*, vol. 28, pp. 139–144, jan 2005.
53. G. Huntington, D. Benson, and A. Rao, "A comparison of accuracy and computational efficiency of three pseudospectral methods," *AIAA Guidance, Navigation and Control Conference and Exhibit*, 2007.
54. A. V. Ivanyukhin and V. G. Petukhov, "The thrust minimization problem and its applications," *Cosmic Research*, vol. 53, pp. 300–310, jul 2015.
55. M. Vasile, P. D. Pascale, and S. Casotto, "On the optimality of a shape-based approach based on pseudo-equinoctial elements," *Acta Astronautica*, vol. 61, pp. 286–297, jun 2007.
56. E. Taheri and O. Abdelkhalik, "Initial three-dimensional low-thrust trajectory design," *Advances in Space Research*, vol. 57, no. 3, pp. 889 – 903, 2016.
57. Y. Ulybyshev, "Trajectory optimization for spacecraft proximity operations with constraints," in *AIAA Guidance, Navigation, and Control Conference*, pp. 1–19, American Institute of Aeronautics and Astronautics, aug 2011.
58. Y. P. Ulybyshev, "Optimization of low-thrust orbit transfers with constraints," *Cosmic Research*, vol. 50, pp. 376–390, sep 2012.
59. L. W. Neustadt, "A general theory of minimum-fuel space trajectories," *Journal of the Society for Industrial and Applied Mathematics, Series A: Control*, vol. 3, no. 2, pp. 317–356, 1965.
60. Y.-Z. Luo and G.-J. Tang, "Spacecraft optimal rendezvous controller design using simulated annealing," *Aerospace Science and Technology*, vol. 9, pp. 732–737, nov 2005.
61. S. Li, R. Mehra, R. Smith, and R. Beard, "Multi-spacecraft trajectory optimization and control using genetic algorithm techniques," in *2000 IEEE Aerospace Conference. Proceedings (Cat. No.00TH8484)*, pp. 99–108, IEEE, 2000.
62. M. Vasile and F. Zuiani, "Multi-agent collaborative search: an agent-based memetic multi-objective optimization algorithm applied to space trajectory design," *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, vol. 225, pp. 1211–1227, sep 2011.
63. B. M. Shippey, *Trajectory optimization using collocation and evolutionary programming for constrained nonlinear dynamical systems.* PhD thesis, The University of Texas at Arlington, 2008.
64. K. Miettinen and M. M. Makela, "On scalarizing functions in multiobjective optimization," *OR Spectrum*, vol. 24, pp. 193–213, may 2002.
65. J. W. Sales Jr, "Trajectory optimization for spacecraft collision avoidance," tech. rep., Air Force Institute of Technology, 2013.
66. A. Shirazi, "Multi-objective optimization of orbit transfer trajectory using imperialist competitive algorithm," in *2017 IEEE Aerospace Conference*, pp. 1–14, March 2017.
67. A. Bolle and C. Circi, "A hybrid, self-adjusting search algorithm for optimal space trajectory design," *Advances in Space Research*, vol. 50, no. 4, pp. 471 – 488, 2012.
68. R. Vinter, *Optimal control.* Springer Science & Business Media, 2010.

69. A. A. Quarta and G. Mengali, "Semi-analytical method for the analysis of solar sail heliocentric orbit raising," *Journal of Guidance, Control, and Dynamics*, vol. 35, pp. 330–335, jan 2012.
70. O. von Stryk and R. Bulirsch, "Direct and indirect methods for trajectory optimization," *Annals of Operations Research*, vol. 37, pp. 357–373, Dec 1992.
71. I. M. Ross, "A historical introduction to the convector mapping principle," in *Proceedings of Astrodynamics Specialists Conference*, pp. 1–21, Naval Postgraduate School (US), 2005.
72. I. M. Ross and F. Fahroo, "A pseudospectral transformation of the convectors of optimal control systems," *IFAC Proceedings Volumes*, vol. 34, no. 13, pp. 543–548, 2001.
73. I. M. Ross and F. Fahroo, "A unified computational framework for real-time optimal control," in *Decision and Control, 2003. Proceedings. 42nd IEEE Conference on*, vol. 3, pp. 2210–2215, IEEE, 2003.
74. M. Ross and F. Fahroo, "Discrete verification of necessary conditions for switched nonlinear optimal control systems," in *American Control Conference, 2004. Proceedings of the 2004*, vol. 2, pp. 1610–1615, IEEE, 2004.
75. F. L. Lewis, D. L. Vrabie, and V. L. Syrmos, *Optimal Control*. John Wiley & Sons, Inc., jan 2012.
76. S. da Silva Fernandes, F. das Chagas Carvalho, and R. V. de Moraes, "Optimal low-thrust transfers between coplanar orbits with small eccentricities," *Computational and Applied Mathematics*, vol. 35, pp. 803–816, jul 2015.
77. A. V. Rao, "A survey of numerical methods for optimal control," *Advances in the Astronautical Sciences*, vol. 135, pp. 497 – 528, 2009.
78. J. Z. Ben-Asher, *Optimal Control Theory with Aerospace Applications*. American Institute of Aeronautics and Astronautics, jan 2010.
79. E. Taheri and O. Abdelkhalik, "Shape based approximation of constrained low-thrust space trajectories using fourier series," *Journal of Spacecraft and Rockets*, vol. 49, pp. 535–546, may 2012.
80. E. Taheri and O. O. Abdelkhalik, "Constraint low-thrust trajectory planning in three-body dynamic models: Fourier series approach," in *AIAA/AAS Astrodynamics Specialist Conference*, American Institute of Aeronautics and Astronautics, Aug. 2014.
81. N. P. Nurre and E. Taheri, "Multiple gravity-assist low-thrust trajectory design using finite fourier series," in *AIAA/AAS Astrodynamics Specialist Conference*, American Institute of Aeronautics and Astronautics, Aug. 2020.
82. D. P. Bertsekas, *Dynamic Programming and Optimal Control*, vol. 2. Athena Scientific, 4 ed., 2016.
83. A. Bressan and G. Facchi, "Trajectories of differential inclusions with state constraints," *Journal of Differential Equations*, vol. 250, pp. 2267–2281, feb 2011.
84. M. Diehl, H. G. Bock, H. Diedam, and P.-B. Wieber, "Fast direct multiple shooting algorithms for optimal robot control," in *Fast motions in biomechanics and robotics*, pp. 65–93, Springer, 2006.
85. D. A. Benson, G. T. Huntington, T. P. Thorvaldsen, and A. V. Rao, "Direct trajectory optimization and costate estimation via an orthogonal collocation method," *Journal of Guidance, Control, and Dynamics*, vol. 29, no. 6, pp. 1435–1440, 2006.

86. A. V. Rao, D. A. Benson, C. Darby, M. A. Patterson, C. Francolin, I. Sanders, and G. T. Huntington, "Algorithm 902: GPOPS, A MATLAB software for solving multiple-phase optimal control problems using the gauss pseudospectral method," *ACM Trans. Math. Softw.*, vol. 37, pp. 22:1–22:39, Apr. 2010.
87. D. Garg, M. A. Patterson, C. Francolin, C. L. Darby, G. T. Huntington, W. W. Hager, and A. V. Rao, "Direct trajectory optimization and costate estimation of finite-horizon and infinite-horizon optimal control problems using a radau pseudospectral method," *Computational Optimization and Applications*, vol. 49, pp. 335–358, Jun 2011.
88. H. Hou, W. Hager, and A. Rao, "Convergence of a gauss pseudospectral method for optimal control," *AIAA Guidance, Navigation, and Control Conference*, 2012.
89. D. Garg, M. Patterson, W. W. Hager, A. V. Rao, D. A. Benson, and G. T. Huntington, "A unified framework for the numerical solution of optimal control problems using pseudospectral methods," *Automatica*, vol. 46, no. 11, pp. 1843 – 1851, 2010.
90. D. Garg, W. W. Hager, and A. V. Rao, "Pseudospectral methods for solving infinite-horizon optimal control problems," *Automatica*, vol. 47, pp. 829–837, apr 2011.
91. D. Garg, M. A. Patterson, W. W. Hager, A. V. Rao, D. Benson, and G. T. Huntington, "An overview of three pseudospectral methods for the numerical solution of optimal control problems," *Advances in the Astronautical Sciences*, pp. 475 – 487, 2009.
92. X. Guo and M. Zhu, "Direct trajectory optimization based on a mapped chebyshev pseudospectral method," *Chinese Journal of Aeronautics*, vol. 26, no. 2, pp. 401 – 412, 2013.
93. Q. Gong, F. Fahroo, I. M. Ross, *et al.*, "Spectral algorithm for pseudospectral methods in optimal control," *Journal of Guidance Control and Dynamics*, vol. 31, no. 3, pp. 460–471, 2008.
94. P. Williams, "Hermite-legendre-gauss-lobatto direct transcription in trajectory optimization," *Journal of Guidance, Control, and Dynamics*, vol. 32, no. 4, pp. 1392–1395, 2009.
95. I. M. Ross and M. Karpenko, "A review of pseudospectral optimal control: From theory to flight," *Annual Reviews in Control*, vol. 36, no. 2, pp. 182 – 197, 2012.
96. F. Topputo and C. Zhang, "Survey of direct transcription for low-thrust space trajectory optimization with applications," *Abstract and Applied Analysis*, vol. 2014, pp. 1–15, 2014.
97. A. Engelsone and S. L. Campbell, "Adjoint estimation using direct transcription multipliers: compressed trapezoidal method," *Optimization and Engineering*, vol. 9, no. 3, pp. 291–305, 2008.
98. A. Miele and A. Salvetti, *Applied Mathematics in Aerospace Science and Engineering*. Springer, 2014.
99. L. S. Breger and J. P. How, "Safe trajectories for autonomous rendezvous of spacecraft," *Journal of Guidance, Control, and Dynamics*, vol. 31, pp. 1478–1489, sep 2008.
100. S. Campagnola, P. Skerritt, and R. P. Russell, "Flybys in the planar, circular, restricted, three-body problem," *Celestial Mechanics and Dynamical Astronomy*, vol. 113, pp. 343–368, Jul 2012.

101. P. E. Gill, W. Murray, and M. A. Saunders, "SNOPT: An SQP algorithm for large-scale constrained optimization," *SIAM review*, vol. 47, no. 1, pp. 99–131, 2005.
102. P. Spellucci, "DONLP2 short users guide," *Technische Universitt Darmstadt*, 1999.
103. R. Fletcher and S. Leyffer, "User manual for filterSQP," *Numerical Analysis Report NA/181, Department of Mathematics, University of Dundee, Dundee, Scotland*, p. 35, 1998.
104. R. A. Bartlett and L. T. Biegler, "rSQP++: An object-oriented framework for successive quadratic programming," in *Large-Scale PDE-Constrained Optimization*, pp. 316–330, Springer, 2003.
105. R. H. Byrd, J. Nocedal, and R. A. Waltz, "KNITRO: An integrated package for nonlinear optimization," in *Large-scale nonlinear optimization*, pp. 35–59, Springer, 2006.
106. "NEOS server for optimization." (Online; accessed 30-May-2018).
107. "AIMMS: Prescriptive analytics and supply chain management." (Online; accessed 30-May-2018).
108. "AMPL: Streamlined modeling for real optimization." (Online; accessed 30-May-2018).
109. "GAMS: The general algebraic modeling system." (Online; accessed 30-May-2018).
110. D. P. Bertsekas, *Nonlinear programming*. Athena scientific Belmont, 2016.
111. H. Seywald, R. R. Kumar, and S. M. Deshpande, "Genetic algorithm approach for optimal control problems with linearly appearing controls," *Journal of Guidance, Control, and Dynamics*, vol. 18, no. 1, pp. 177–182, 1995.
112. S. Voß, S. Martello, I. H. Osman, and C. Roucairol, eds., *Meta-Heuristics: Advances and Trends in Local Search Paradigms for Optimization*. Springer US, 1998.
113. C. Blum and A. Roli, "Metaheuristics in combinatorial optimization: Overview and conceptual comparison," *ACM computing surveys*, vol. 35, pp. 268–308, Sept. 2003.
114. R. Popa, *Genetic Algorithms in Applications*. InTech, 2012.
115. O. Kramer, *Machine Learning for Evolution Strategies*. Springer International Publishing, 2016.
116. R. Riolo, E. Vladislavleva, M. D. Ritchie, and J. H. Moore, *Genetic Programming Theory and Practice X*. Springer, 2013.
117. E. P. Konstantinos and N. V. Michael, "Introduction," in *Particle Swarm Optimization and Intelligence*, pp. 1–24, IGI Global, 2010.
118. N. Xiong, D. Molina, M. L. Ortiz, and F. Herrera, "A walk into metaheuristics for engineering optimization: Principles, methods and recent trends," *International Journal of Computational Intelligence Systems*, vol. 8, no. 4, pp. 606–636, 2015.
119. M. A. Muñoz, Y. Sun, M. Kirley, and S. K. Halgamuge, "Algorithm selection for black-box continuous optimization problems: A survey on methods and challenges," *Information Sciences*, vol. 317, pp. 224 – 245, 2015.
120. I. F. Jr., X. Yang, I. Fister, J. Brest, and D. Fister, "A brief review of nature-inspired algorithms for optimization," *CoRR*, vol. abs/1307.4186, 2013.
121. A. Shirazi and A. Mazinan, "Mathematical modeling of spacecraft guidance and control system in 3d space orbit transfer mission," *Computational and Applied Mathematics*, vol. 35, no. 3, pp. 865–879, 2016.

122. H.-D. Kim, O.-C. Jung, and H. Bang, "A computational approach to reduce the revisit time using a genetic algorithm," in *2007 International Conference on Control, Automation and Systems*, pp. 184–189, Oct 2007.
123. A. Shirazi, "Analysis of a hybrid genetic simulated annealing strategy applied in multi-objective optimization of orbital maneuvers," *IEEE Aerospace and Electronic Systems Magazine*, vol. 32, pp. 6–22, January 2017.
124. A. Shirazi, "Trajectory optimization of spacecraft high-thrust orbit transfer using a modified evolutionary algorithm," *Engineering Optimization*, vol. 48, no. 10, pp. 1639–1657, 2016.
125. M. Vasile, E. Minisci, and M. Locatelli, "An inflationary differential evolution algorithm for space trajectory optimization," *IEEE Transactions on Evolutionary Computation*, vol. 15, pp. 267–281, April 2011.
126. B. Addis, A. Cassioli, M. Locatelli, and F. Schoen, "A global optimization method for the design of space trajectories," *Computational Optimization and Applications*, vol. 48, pp. 635–652, Apr 2011.
127. J. A. Englander, B. A. Conway, and T. Williams, "Automated mission planning via evolutionary algorithms," *Journal of Guidance, Control, and Dynamics*, vol. 35, no. 6, pp. 1878–1887, 2012.
128. H. Lei, B. Xu, and L. Zhang, "Trajectory design for a rendezvous mission to Earth's trojan asteroid 2010TK7," *Advances in Space Research*, vol. 60, pp. 2505–2517, Dec. 2017.
129. M. Pontani, "Optimal low-thrust hyperbolic rendezvous for Earth-Mars missions," *Acta Astronautica*, Feb. 2019.
130. J. Li, "Revisiting the fuel-optimal four-impulse rendezvous problem near circular orbits," *Advances in Space Research*, vol. 60, pp. 2181–2194, Nov. 2017.
131. K. Kitamura, K. Yamada, and T. Shima, "Minimum energy coplanar orbit transfer of geostationary spacecraft using time-averaged hamiltonian," *Acta Astronautica*, vol. 160, pp. 270–279, July 2019.
132. H. Chen, C. Han, Y. Rao, J. Yin, and X. Sun, "Algorithm of relative lambert transfer based on relative orbital elements," *Journal of Guidance, Control, and Dynamics*, pp. 1–10, jan 2019.
133. A. Shirazi, J. Ceberio, and J. A. Lozano, "Optimal multi-impulse space rendezvous considering limited impulse using a discretized lambert problem combined with evolutionary algorithms," in *8th European Conference for Aeronautics and Space Sciences*, 2019.
134. Z. Zheng, J. Guo, and E. Gill, "Distributed onboard mission planning for multi-satellite systems," *Aerospace Science and Technology*, vol. 89, pp. 111–122, June 2019.
135. M. Pontani and B. A. Conway, "Particle swarm optimization applied to space trajectories," *Journal of Guidance, Control, and Dynamics*, vol. 33, pp. 1429–1441, sep 2010.
136. A. A. Quarta and G. Mengali, "Simple solution to optimal cotangential transfer between coplanar elliptic orbits," *Acta Astronautica*, vol. 155, pp. 247–254, Feb. 2019.
137. J. Zhang, G. jin Tang, and Y. zhong Luo, "Optimization of an orbital long-duration rendezvous mission," *Aerospace Science and Technology*, vol. 58, pp. 482–489, Nov. 2016.
138. M. S. Mohammadi and A. Naghash, "Robust optimization of impulsive orbit transfers under actuation uncertainties," *Aerospace Science and Technology*, vol. 85, pp. 246–258, Feb. 2019.

139. H. Ma and S. Xu, "Optimization of bounded low-thrust rendezvous with terminal constraints by interval analysis," *Aerospace Science and Technology*, vol. 79, pp. 58–69, Aug. 2018.
140. K. Lee, C. Park, and S.-Y. Park, "Near-optimal continuous control for spacecraft collision avoidance maneuvers via generating functions," *Aerospace Science and Technology*, vol. 62, pp. 65–74, Mar. 2017.
141. G. Zhang, D. Zhou, D. Mortari, and T. A. Henderson, "Analytical study of tangent orbit and conditions for its solution existence," *Journal of Guidance, Control, and Dynamics*, vol. 35, no. 1, pp. 186–194, 2012.
142. G. Zhang, G. Ma, and D. Li, "Two-impulse transfer between coplanar elliptic orbits using along-track thrust," *Celestial Mechanics and Dynamical Astronomy*, vol. 121, no. 3, pp. 261–274, 2015.
143. S. Mitani and H. Yamakawa, "Continuous-thrust transfer with control magnitude and direction constraints using smoothing techniques," *Journal of Guidance, Control, and Dynamics*, vol. 36, no. 1, pp. 163–174, 2012.
144. M. Avendaño, V. Martín-Molina, J. Martín-Morales, and J. Ortigas-Galindo, "Algebraic approach to the minimum-cost multi-impulse orbit-transfer problem," *Journal of Guidance, Control, and Dynamics*, vol. 39, pp. 1734–1743, aug 2016.
145. G. Zhang, X. Cao, and D. Zhou, "Two-impulse cotangent rendezvous between coplanar elliptic and hyperbolic orbits," *Journal of Guidance, Control, and Dynamics*, vol. 37, pp. 964–970, may 2014.
146. P. V. Arlulkar and S. D. Naik, "Dynamical approach for optimal two-impulse rendezvous between elliptic orbits," *Journal of Guidance, Control, and Dynamics*, vol. 37, pp. 1008–1015, may 2014.
147. Z. Yang, Y.-Z. Luo, J. Zhang, and G.-J. Tang, "Homotopic perturbed lambert algorithm for long-duration rendezvous optimization," *Journal of Guidance, Control, and Dynamics*, vol. 38, pp. 2215–2223, nov 2015.
148. E. Taheri and J. L. Junkins, "Generic smoothing for optimal bang-off-bang spacecraft maneuvers," *Journal of Guidance, Control, and Dynamics*, vol. 41, no. 11, pp. 2470–2475, 2018.
149. J. Shimoun, E. Taheri, I. Kolmanovsky, and A. Girard, "A study on GPU-enabled lambert's problem solution for space targeting missions," in *2018 Annual American Control Conference (ACC)*, pp. 664–669, IEEE, June 2018.
150. C. F. Gauss, *Theory of the Motion of the Heavenly Bodies Moving about the Sun in Conic Sections: A Translation of Gauss's" Theoria Motus." With an Appendix*. Little, Brown and Company, 1857.
151. G. Avanzini, "A simple lambert algorithm," *Journal of Guidance, Control, and Dynamics*, vol. 31, pp. 1587–1594, nov 2008.
152. D. Izzo, "Revisiting lambert's problem," *Celestial Mechanics and Dynamical Astronomy*, vol. 121, pp. 1–15, oct 2014.
153. S. L. Nelson and P. Zarchan, "Alternative approach to the solution of lambert's problem," *Journal of Guidance, Control, and Dynamics*, vol. 15, no. 4, pp. 1003–1009, 1992.
154. Y.-Z. Luo, G.-J. Tang, Z.-G. Wang, and H.-Y. Li, "Optimization of perturbed and constrained fuel-optimal impulsive rendezvous using a hybrid approach," *Engineering Optimization*, vol. 38, no. 8, pp. 959–973, 2006.
155. S. A. Darani and O. Abdelkhalik, "Convergence analysis of hidden genes genetic algorithms in space trajectory optimization," *Journal of Aerospace Information Systems*, vol. 15, pp. 228–238, apr 2018.

156. M. Taherkhani and R. Safabakhsh, "A novel stability-based adaptive inertia weight for particle swarm optimization," *Applied Soft Computing*, vol. 38, pp. 281–295, jan 2016.
157. J. Nocedal and S. Wright, *Numerical optimization*. Springer Science & Business Media, 2006.
158. M. R. Bonyadi, Z. Michalewicz, and X. Li, "An analysis of the velocity updating rule of the particle swarm optimization algorithm," *Journal of Heuristics*, vol. 20, no. 4, pp. 417–452, 2014.
159. J. Barrera, O. Álvarez-Bajo, J. J. Flores, and C. A. Coello Coello, "Limiting the velocity in the particle swarm optimization algorithm," *Computación y Sistemas*, vol. 20, no. 4, pp. 635–645, 2016.
160. M. Alhussein and S. I. Haider, "Improved particle swarm optimization based on velocity clamping and particle penalization," in *2015 3rd International Conference on Artificial Intelligence, Modelling and Simulation (AIMS)*, pp. 61–64, IEEE, 2015.
161. V. G. Petukhov, "Application of the angular independent variable and its regularizing transformation in the problems of optimizing low-thrust trajectories," *Cosmic Research*, vol. 57, pp. 351–363, Sept. 2019.
162. E. Taheri, "Composite smooth control method for low-thrust trajectory design: Variable specific impulse engine," in *AIAA Scitech 2020 Forum*, p. 2184, 2020.
163. R. Woollands and E. Taheri, "Optimal low-thrust gravity perturbed orbit transfers with shadow constraints," in *The 2019 AAS/AIAA Astrodynamics Specialist Conference*, (Portland, Maine), 2019.
164. K. M. Malan, J. F. Oberholzer, and A. P. Engelbrecht, "Characterising constrained continuous optimisation problems," in *2015 IEEE Congress on Evolutionary Computation (CEC)*, pp. 1351–1358, IEEE, 2015.
165. M. Zuo, G. Dai, L. Peng, M. Wang, Z. Liu, and C. Chen, "A case learning-based differential evolution algorithm for global optimization of interplanetary trajectory design," *Applied Soft Computing*, vol. 94, p. 106451, Sept. 2020.
166. X. Tian and Y. Jia, "Analytical solutions to the matrix inequalities in the robust control scheme based on implicit lyapunov function for spacecraft rendezvous on elliptical orbit," *IET Control Theory & Applications*, vol. 11, pp. 1983–1991, Aug. 2017.
167. J. Zhang, J. D. Biggs, D. Ye, and Z. Sun, "Finite-time attitude set-point tracking for thrust-vectoring spacecraft rendezvous," *Aerospace Science and Technology*, vol. 96, p. 105588, Jan. 2020.
168. R. Neves and J. P. Sanchez Cuartielles, "Gauss variational equations for low-thrust optimal control problems in low-energy regimes," in *Proceedings of the 69th Int. Astronautical Congress*, International Astronautical Federation, 2018.
169. I. Elliott, C. Sullivan, N. Bosanac, J. R. Stuart, and F. Alibay, "Designing low-thrust trajectories for a smallsat mission to Sun-Earth L5," *Journal of Guidance, Control, and Dynamics*, pp. 1–11, July 2020.
170. D. H. Ellison, B. A. Conway, J. A. Englander, and M. T. Ozimek, "Analytic gradient computation for bounded-impulse trajectory models using two-sided shooting," *Journal of Guidance, Control, and Dynamics*, vol. 41, pp. 1449–1462, July 2018.
171. A. Fossà and C. Bettanini, "Optimal rendezvous trajectory between sample return orbiter and orbiting sample container in a Mars sample return mission," *Acta Astronautica*, vol. 171, pp. 31–41, June 2020.

172. C. Greco, M. D. Carlo, M. Vasile, and R. Epenoy, "Direct multiple shooting transcription with polynomial algebra for optimal control problems under uncertainty," *Acta Astronautica*, vol. 170, pp. 224–234, May 2020.
173. C. A. C. Coello, "Constraint-handling techniques used with evolutionary algorithms," in *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, pp. 773–799, ACM, 2018.
174. A. Panda and S. Pani, "A symbiotic organisms search algorithm with adaptive penalty function to solve multi-objective constrained optimization problems," *Applied Soft Computing*, vol. 46, pp. 344–360, 2016.
175. B. Li, K. Tang, J. Li, and X. Yao, "Stochastic ranking algorithm for many-objective optimization based on multiple indicators," *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 6, pp. 924–938, 2016.
176. R. Chai, A. Savvaris, A. Tsourdos, S. Chai, and Y. Xia, "Unified multiobjective optimization scheme for aeroassisted vehicle trajectory planning," *Journal of Guidance, Control, and Dynamics*, vol. 41, pp. 1521–1530, July 2018.
177. M. Vasile and L. Ricciardi, "A direct memetic approach to the solution of multi-objective optimal control problems," in *2016 IEEE Symposium Series on Computational Intelligence (SSCI)*, IEEE, Dec. 2016.
178. F. A. Zotes and M. S. Peñas, "Particle swarm optimisation of interplanetary trajectories from Earth to Jupiter and Saturn," *Engineering Applications of Artificial Intelligence*, vol. 25, no. 1, pp. 189–199, 2012.
179. K. Li, R. Chen, G. Fu, and X. Yao, "Two-archive evolutionary algorithm for constrained multiobjective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 2, pp. 303–315, 2018.
180. P. Larrañaga and J. A. Lozano, *Estimation of distribution algorithms: A new tool for evolutionary computation*. Springer, 2001.
181. J. Ceberio, A. Mendiburu, and J. A. Lozano, "Distance-based exponential probability models on constrained combinatorial optimization problems," in *Proc. Gen. Evol. Comp. Conf. GECCO 18*, ACM Press, 2018.
182. J. Ceberio, A. Mendiburu, and J. A. Lozano, "A square lattice probability model for optimising the graph partitioning problem," in *2017 IEEE Congress on Evolutionary Computation (CEC)*, IEEE, June 2017.
183. J. Sullivan, S. Grimberg, and S. D'Amico, "Comprehensive survey and assessment of spacecraft relative motion dynamics models," *Journal of Guidance, Control, and Dynamics*, vol. 40, pp. 1837–1859, Aug. 2017.
184. K. Yamanaka and F. Ankersen, "New state transition matrix for relative motion on an arbitrary elliptical orbit," *Journal of Guidance, Control, and Dynamics*, vol. 25, pp. 60–66, Jan. 2002.
185. Y. Wang and H. Ji, "Integrated relative position and attitude control for spacecraft rendezvous with ISS and finite-time convergence," *Aerospace Science and Technology*, vol. 85, pp. 234–245, Feb. 2019.
186. P. R. A. Gilz, M. Joldes, C. Louembet, and F. Camps, "Stable model predictive strategy for rendezvous hovering phases allowing for control saturation," *Journal of Guidance, Control, and Dynamics*, vol. 42, pp. 1658–1675, Aug. 2019.
187. K. Subbarao and S. Welsh, "Nonlinear control of motion synchronization for satellite proximity operations," *Journal of Guidance, Control, and Dynamics*, vol. 31, pp. 1284–1294, Sept. 2008.
188. J. R. Rice, *Numerical Methods in Software and Analysis*. Elsevier, 2014.

189. F. N. Fritsch and R. E. Carlson, "Monotone piecewise cubic interpolation," *SIAM Journal on Numerical Analysis*, vol. 17, no. 2, pp. 238–246, 1980.
190. C. De Boor, C. De Boor, E.-U. Mathématicien, C. De Boor, and C. De Boor, *A practical guide to splines*, vol. 27. springer-verlag New York, 1978.
191. E. Catmull and R. Rom, "A class of local interpolating splines," in *Computer Aided Geometric Design*, pp. 317–326, Elsevier, 1974.
192. G. Behforooz, "The not-a-knot piecewise interpolatory cubic polynomial," *Applied Mathematics and Computation*, vol. 52, pp. 29–35, Nov. 1992.
193. D. Arthur and S. Vassilvitskii, "k-means++: The advantages of careful seeding," tech. rep., Stanford, 2006.
194. C. A. C. Coello, "Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art," *Computer methods in applied mechanics and engineering*, vol. 191, no. 11-12, pp. 1245–1287, 2002.
195. V. Hodge and J. Austin, "A survey of outlier detection methodologies," *Artificial intelligence review*, vol. 22, no. 2, pp. 85–126, 2004.
196. T. P. Runarsson and X. Yao, "Stochastic ranking for constrained evolutionary optimization," *IEEE Trans. Evol. Comp.*, vol. 4, no. 3, pp. 284–294, 2000.
197. A. Kumar, G. Wu, M. Z. Ali, R. Mallipeddi, P. N. Suganthan, and S. Das, "A test-suite of non-convex constrained optimization problems from the real-world and some baseline results," *Swarm and Evolutionary Computation*, p. 100693, 2020.
198. M. Andersson, S. Bandaru, A. H. Ng, and A. Syberfeldt, "Parameter tuned CMA-ES on the CEC 15 expensive problems," in *IEEE Congress on Evolutionary Computation*, IEEE, May 2015.
199. S. Rodrigues, P. Bauer, and P. A. Bosman, "A novel population-based multi-objective cma-es and the impact of different constraint handling techniques," in *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, pp. 991–998, 2014.
200. A. Kumar, S. Das, and I. Zelinka, "A self-adaptive spherical search algorithm for real-world constrained optimization problems," in *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, ACM, July 2020.
201. J. Gurrola-Ramos, A. Hernandez-Aguirre, and O. Dalmau-Cedeno, "COLSHADE for real-world single-objective constrained optimization problems," in *IEEE Congress on Evolutionary Computation*, IEEE, July 2020.
202. A. Kumar, S. Das, and I. Zelinka, "A modified covariance matrix adaptation evolution strategy for real-world constrained optimization problems," in *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, ACM, July 2020.
203. K. M. Sallam, S. M. Elsayed, R. K. Chakrabortty, and M. J. Ryan, "Multi-operator differential evolution algorithm for solving real-world constrained optimization problems," in *IEEE Congress on Evolutionary Computation*, IEEE, July 2020.
204. M. Hellwig and H.-G. Beyer, "A modified matrix adaptation evolution strategy with restarts for constrained real-world problems," in *IEEE Congress on Evolutionary Computation*, IEEE, July 2020.
205. Z. Fan, Y. Fang, W. Li, Y. Yuan, Z. Wang, and X. Bian, "LSHADE44 with an improved constraint-handling method for solving constrained single-objective optimization problems," in *IEEE Congress on Evolutionary Computation*, IEEE, July 2018.

206. Y. Huang and Y. Jia, "Robust adaptive fixed-time tracking control of 6-DOF spacecraft fly-around mission for noncooperative target," *International Journal of Robust and Nonlinear Control*, vol. 28, pp. 2598–2618, Jan. 2018.
207. L. Feruglio and S. Corpino, "Neural networks to increase the autonomy of interplanetary nanosatellite missions," *Robotics and Autonomous Systems*, vol. 93, pp. 52–60, July 2017.
208. B. Davoudi, E. Taheri, K. Duraisamy, B. Jayaraman, and I. Kolmanovsky, "Quad-rotor flight simulation in realistic atmospheric conditions," *AIAA Journal*, vol. 58, no. 5, pp. 1992–2004, 2020.
209. A. Shirazi and M. Mirshams, "Design and performance simulation of a satellite momentum exchange actuator," *Australian Journal of Mechanical Engineering*, vol. 14, no. 1, pp. 1–9, 2016.
210. B. Jones, E. Delande, E. Zucchelli, and M. Jah, "Multi-fidelity orbit uncertainty propagation with systematic errors," *amos*, p. 14, 2019.
211. M. M. Teshnizi and A. Shirazi, "Attitude estimation and sensor identification utilizing nonlinear filters based on a low-cost mems magnetometer and sun sensor," *IEEE Aerospace and Electronic Systems Magazine*, vol. 30, no. 12, pp. 20–33, 2015.
212. J. A. Dominguez, E. Victor, A. R. Urbina, and A. L. Vasquez, "Comprehensive software simulation on ground power supply for launch pads and processing facilities at NASA kennedy space center," *IEEE Aerospace and Electronic Systems Magazine*, vol. 34, pp. 4–11, Aug. 2019.
213. A. Graphics, "Incorporated (agi). stk user's manual [k]. version5. 0.1. extort, pa, usa: Analytical graphics," *Inc (AGI)*, 2002.
214. D. J. Conway and S. P. Hughes, "The general mission analysis tool (GMAT): Current features and adding custom functionality," in *4th International Conference on Astrodynamical Tools and Techniques*, pp. 1–8, 2010.
215. V. Carrara, "An open source satellite attitude and orbit simulator toolbox for matlab," in *Proceedings of the 17th International Symposium on Dynamic Problems of Mechanics*, pp. 1–12, 2015.
216. D. Brouwer and G. M. Clemence, *Methods of celestial mechanics*. Academic, New York, 1961.
217. S. Habibkhah, J. Arasi, and H. Bolandi, "SPACSSIM: Simulation and analysis software for mathematical modeling of satellite position and attitude control system," *Computing in Science & Engineering*, pp. 1–1, 2017.
218. O. R. Fernandez, J. Utzmann, and U. Hugentobler, "SPOOK - a comprehensive space surveillance and tracking analysis tool," *Acta Astronautica*, feb 2018.
219. A. Shirazi, "HOMA - Online Space Orbit Simulator," 2014. Accessed: 2020-12-14.
220. R. A. Broucke and P. J. Cefola, "On the equinoctial orbit elements," *Celestial Mechanics*, vol. 5, no. 3, pp. 303–310, 1972.
221. D. A. Vallado and P. J. Cefola, "Two-line element sets–practice and use," in *63rd International Astronautical Congress, Naples, Italy*, pp. 1–14, 2012.
222. A. Shirazi and M. Mirshams, "Pyramidal reaction wheel arrangement optimization of satellite attitude control subsystem for minimizing power consumption," *International Journal of Aeronautical and Space Sciences*, vol. 15, no. 2, pp. 190–198, 2014.
223. F. R. Hoots and R. L. Roehrich, "Models for propagation of NORAD element sets," tech. rep., Aerospace defense command Peterson AFB Co-office of Astrodynamics, 1980.

224. D. Brouwer, "Solution of the problem of artificial satellite theory without drag," *The Astronomical Journal*, vol. 64, p. 378, nov 1959.
225. Y. Kozai, "The motion of a close Earth satellite," *The Astronomical Journal*, vol. 64, p. 367, nov 1959.
226. M. Lane, P. Fitzpatrick, and J. Murphy, "On the representation of air density in satellite deceleration equations by power functions with integral exponents," tech. rep., Air Proving Ground Center EGLIN AFB FL, 1962.
227. D. Vallado and P. Crawford, "SGP4 orbit determination," in *AIAA/AAS Astrodynamics Specialist Conference and Exhibit*, p. 6770, 2008.
228. W. M. Kaula, *Theory of satellite geodesy: applications of satellites to geodesy*. Courier Corporation, 2013.
229. J. M. Wahr, "Body tides on an elliptical, rotating, elastic and oceanless Earth," *Geophysical Journal International*, vol. 64, no. 3, pp. 677–703, 1981.
230. R. Eanes, "Earth and ocean tide effects on Lageos and Starlette," in *Proceedings of the Ninth International Symposium on Earth Tides*, pp. 1–17, E. Sckweizerbart'sche Verlagabuchhandlung, 1983.
231. K. Lambeck, *The Earth's variable rotation: geophysical causes and consequences*. Cambridge University Press, 2005.
232. E. Standish, J. Planetary, and L. Ephemerides, "DE405/LE405, Memo IOM 312," tech. rep., F-98-048, 1998.
233. C. Huang, J. Ries, B. Tapley, and M. Watkins, "Relativistic effects for near-Earth satellite orbit determination," *Celestial Mechanics and Dynamical Astronomy*, vol. 48, no. 2, pp. 167–185, 1990.
234. J. Ries, C. Huang, and M. Watkins, "Effect of general relativity on a near-Earth satellite in the geocentric and barycentric reference frames," *Physical review letters*, vol. 61, no. 8, p. 903, 1988.
235. B. Schutz and B. Tapley, "Orbit accuracy assessment for Seasat," *Journal of the Astronautical Sciences*, vol. 28, pp. 371–390, 1980.
236. B. Tapley and J. Ries, "Orbit determination requirements for TOPEX," in *Astrodynamics 1987*, pp. 321–338, 1988.
237. P. Knocke, J. Ries, and B. Tapley, "Earth radiation pressure effects on satellites," in *Astrodynamics Conference*, p. 4292, 1988.
238. H. Fliegel, T. Gallini, and E. Swift, "Global positioning system radiation force model for geodetic applications," *Journal of Geophysical Research: Solid Earth*, vol. 97, no. B1, pp. 559–568, 1992.
239. A. Shirazi and M. Mirshams, "Software development for satellite thermal design," in *Applied Mechanics and Materials*, vol. 390, pp. 703–707, Trans Tech Publ, 2013.
240. A. Shirazi, J. Ceberio, and J. A. Lozano, "An evolutionary discretized lambert approach for optimal long-range rendezvous considering impulse limit," *Aerospace Science and Technology*, p. 105400, 2019.
241. M. Saghamanesh, E. Taheri, and H. Baoyin, "Systematic low-thrust trajectory design to Mars based on a full ephemeris modeling," *Advances in Space Research*, Aug. 2019.
242. A. Shakouri, M. Kiani, and S. H. Pourtakdoust, "Covariance-based multiple-impulse rendezvous design," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 55, pp. 2128–2137, Oct. 2019.
243. T. Kelso, "Celestrak," *Public Domain Satellite Tracking Data, URL: http://celestrak. com/[cited 13 March 2012]*, 2010.